

METALTEX

NEXO

Manual CLP Nexo

Manual de programação – CLP Nexo

Novembro / 2017

Todos os direitos autorais deste documento são reservados a Metaltex.

A reprodução deste documento, total ou parcialmente, não poderá ser feita sem a autorização por escrito da Metaltex.

A Metaltex se reserva no direito, sem qualquer aviso prévio, de alterar as informações deste documento.

Sumário

1. Introdução.....	12
2. Validade deste manual.....	13
3. Instruções de Segurança	13
4. Prescrições de Uso	14
4.1. Instalação / remoção do CLP Nexo	15
4.2. Montagem em trilho DIN	15
4.3. Montagem com Parafuso.....	17
4.4. Conexão da Fonte de Alimentação	18
4.5. Conectando as entradas	19
4.6. Conectando as Saídas.....	22
4.7. Instruções da porta de comunicação.....	24
5. Guia de referência.....	25
5.1. Área de memórias especiais	25
6. NeXo Ladder informações de uso	27
6.1. Menu principal	27
6.1.1. Arquivo.....	27
6.1.2. Editar.....	28
6.1.3. Exibir	29
6.1.4. CLP.....	31
6.1.5. Depurar	32
6.1.6. Ajuda	32
6.2. Barra de Ferramentas	32
6.3. Instruções.....	34
6.3.1. Projeto.....	35
6.3.2. Bloco de dados.....	35
6.3.3. Bloco de Sistema.....	36
6.3.4. Bloco de Programa	40
6.3.5. Símbolo de função	40
6.3.6. Símbolo de variável.....	40
6.3.7. Monitor de valores.....	41

6.3.8.	Referência cruzada.....	41
6.3.9.	Comunicação.....	42
6.4.	Instruções.....	44
	44
6.4.1.	Projeto.....	44
	44
	As instruções serão explicadas detalhadamente na seção de instruções.	44
6.4.2.	Bloco de dados.....	45
6.4.3.	Bloco de Sistema.....	45
6.4.4.	Bloco de dados.....	49
6.4.5.	Bloco de dados.....	49
6.4.6.	Símbolo de variável.....	50
6.4.7.	Monitor de valores.....	51
6.4.8.	Referência cruzada.....	51
6.4.9.	Referência cruzada.....	51
6.4.10.	Editor de Programas	53
6.4.11.	Saída de Informação	54
6.5.	Conceitos de programação	54
6.5.1.	Como o programa funciona	54
6.5.2.	Visão geral endereçamento	55
6.5.3.	Como organizar o programa	56
6.6.	Como usar o Nexo Ladder	56
6.6.1.	Como contruir um novo projeto	56
6.6.2.	Principio de trabalho em Ladder.....	56
6.6.3.	Regras de paralelos e series.....	57
6.6.4.	Entrada de comandos	57
6.6.5.	Como endereçar os elementos.....	59
6.6.6.	Como editar os elementos.....	59
6.6.7.	Como usar o Encontrar / Substituir	60
6.6.8.	Como identificar erros no programa	61
6.6.9.	Como compilar.....	61
6.6.10.	Como salvar o projeto.....	62

6.7.	Comunicação e transferência do programa.....	62
6.7.1.	Configuração de comunicação.....	62
6.7.2.	Download do programa.....	64
6.7.3.	Como corrigir erros de compilação e erros de download.....	65
6.7.4.	Como monitorar e depurar o programa.....	66
6.7.5.	Operação e opções do CLP.....	68
7.	Descrições de instruções Nexo Ladder.....	69
7.1.	Boleana.....	69
7.1.1.	Normalmente aberto e Normalmente fechado.....	70
7.1.2.	Imediatamente o contato NA e imediatamente o contato NF.....	71
7.1.3.	NOT inverte instrução.....	71
7.1.4.	Borda de subida e borda de descida.....	72
7.1.5.	Saída.....	72
7.1.6.	Saída imediata.....	73
7.1.7.	Set e Reset.....	74
7.1.8.	Set e Reset imediato.....	74
7.1.9.	Instrução SR.....	75
7.1.10.	Instrução RS.....	76
7.1.11.	Instrução NOP.....	78
7.2.	Instrução Relógio.....	78
7.2.1.	Lê e grava a hora em tempo real.....	78
7.3.	Comunicação.....	80
7.3.1.	Ler endereço da porta.....	80
7.3.2.	Definir endereço da porta.....	81
7.4.	Comparação.....	81
7.4.1.	Comparação de Byte.....	82
7.4.2.	Comparação de Inteiros.....	83
7.4.3.	Comparação de duplo inteiro.....	84
7.4.4.	Comparação de número Real.....	85
7.4.5.	Comparação de String.....	86
7.5.	Conversores.....	87
7.5.1.	Byte para integer.....	87

7.5.2.	Integer para byte	88
7.5.3.	Integer para double integer	88
7.5.4.	Integer para string	89
7.5.5.	Double integer para integer	91
7.5.6.	Double integer para número real	91
7.5.7.	Double integer para string	92
7.5.8.	Conversão de BCD para integer e integer para BCD	93
7.5.9.	ROUND	94
7.5.10.	TRUNC	95
7.5.11.	Número Real para string	96
7.5.12.	Integer para código ASCII	98
7.5.13.	Double integer para o código ASCII	100
7.5.14.	Número Real para código ASCII	101
7.5.15.	ATH&HTA	104
7.5.16.	String para integer	105
7.5.17.	String para double integer	107
7.5.18.	String para número real	109
7.5.19.	DECO	111
7.5.20.	ENCO	111
7.5.21.	Código de sete segmentos	112
7.6.	Contadores	113
7.6.1.	CTU	114
7.6.2.	CTD	115
7.6.3.	CTUD	116
7.7.	Real	117
7.7.1.	ADD-R&SUB-R	117
7.7.2.	MUL - R&DIV - R	118
7.7.3.	SQRT	119
7.7.4.	SIN	121
7.7.5.	COS	122
7.7.6.	TAN	123
7.7.7.	LN	125

7.7.8.	EXP	126
7.7.9.	PID	127
7.8.	Inteiros	130
7.8.1.	ADD-I&SUB-I	130
7.8.2.	ADD- DI & SUB- DI	131
7.8.3.	MUL & DIV.....	132
7.8.4.	MUL -I & DIV-I	133
7.8.5.	MUL -DI & DIV -DI	134
7.8.6.	INC-B & DEC-B.....	135
7.8.7.	INC-W & DEC-W	136
7.8.8.	INC -DW & DEC -DW	137
7.9.	Interrupção	138
7.9.1.	ENI & DISI	138
7.9.2.	Instrução RETI	139
7.9.3.	ATCH.....	141
7.9.4.	DTCH	142
7.9.5.	Limpar evento de interrupção	143
7.10.	Lógica binária	144
7.10.1.	INV-B	145
7.10.2.	INV-W	145
7.10.3.	INV-DW	146
7.10.4.	WAND-B, WOR -B, WXOR -B	147
7.10.5.	WAND-W, WOR -W, WXOR -W	148
7.10.6.	WAND- DW, WOR -DW, WXOR -DW.....	149
7.11.	Mover	150
7.11.1.	Move Byte	150
7.11.2.	Move Word	151
7.11.3.	Move Double word	152
7.11.4.	Move Número real.....	153
7.11.5.	BLKMOV -B.....	153
7.11.6.	BLKMOV-W	154
7.11.7.	BLKMOV -D.....	155

7.11.8.	SWAP.....	156
7.11.9.	MOV-BIR.....	157
7.11.10.	MOV-BIW	158
7.12.	Controles de Programa	159
7.12.1.	FOR, NEXT	159
7.12.2.	Pular	160
7.12.3.	Relé de controle de sequência.....	161
7.12.4.	Retorno da sub-rotina.....	163
7.12.5.	Fim condicional.....	164
7.12.6.	STOP	165
7.12.7.	Watchdog Reset.....	165
7.12.8.	Diagnosis LED	166
7.13.	Deslocamento / Rotação.....	167
7.13.1.	SHR -B & SHL -B.....	167
7.13.2.	SHR -W & SHL -W	168
7.13.3.	SHR -DW & SHL -DW	169
7.13.4.	ROR -B & ROL -B.....	170
7.13.5.	ROR -W & ROL -W	171
7.13.6.	ROR -DW & ROL -DW	172
7.13.7.	SHRB.....	173
7.14.	String.....	174
7.14.1.	Tamanho da String.....	175
7.14.2.	Copiar String.....	175
7.14.3.	SSTR-CPY	176
7.14.4.	Concatenar String	177
7.14.5.	STR-FIND	178
7.14.6.	Procura o primeiro caractere na string.....	180
7.15.	Tabela.....	181
7.15.1.	Último a entrar é o primeiro a sair	181
7.15.2.	Primeiro a entrar é o primeiro a sair	183
7.15.3.	Adicionar à tabela.....	185
7.15.4.	Preencher memória	186

7.15.5.	Tabela de busca	187
7.16.	Temporizadores	190
7.16.1.	TON	190
7.16.2.	TONR	192
7.16.3.	TOF	193
7.16.4.	Intervalo de tempo inicial	194
7.16.5.	Tempo de intervalo de cálculo.....	195
7.17.	Trem de Pulsos	196
7.17.1.	Saída de Pulsos.....	196
7.17.2.	Módulo de largura de pulso.....	197
7.18.	Driver UART.....	198
7.18.1.	UART-Init.....	198
7.19.	Comunicação Aberta (UART).....	199
7.19.1.	UFP-RCV	200
7.19.2.	UFP-XMT	201
7.19.3.	UFP-QAR.....	202
7.20.	Modbus (UART).....	203
7.20.1.	UMB-03	203
7.20.2.	UMB-04	204
7.20.3.	UMB-06	204
7.20.4.	UMB-16.....	205
7.21.	CAN-Driver	205
7.21.1.	CAN-Init.....	206
7.22.	Free-Port (CAN).....	207
7.22.1.	CFP-Setup.....	207
7.22.2.	CFP-Rcv.....	207
7.22.3.	CFP-Xmt.....	207
7.23.	LCD	207
7.23.1.	LCD-KEY	208
7.23.2.	LCD-PAGE	208
7.23.3.	LCD-EDIT.....	209
7.24.	Sub-rotinas.....	209

7.24.1.	Usando a sub-rotina.....	210
7.24.2.	Usando parâmetros para chamar sub-rotina	210
7.24.3.	Como configurar uma sub-rotina.....	211
7.24.4.	Como chamar uma sub-rotina	212
8.	Área de armazenamento do CLP.....	214
8.1.	Tipos e propriedades da área de armazenamento	214
8.2.	Endereçamento direto e indireto	214
8.3.	Bit, byte, word e double word	217
8.4.	Alcance do endereço da memória	217
8.5.	Tipo de dados.....	219
8.6.	Constante	219
9.	Atribuição e função da área de armazenamento especial SM	220
9.1.	Área de armazenamento especial SM	220
10.	Comunicação Nexo Ladder	221
10.1.	CLP NEXO - Introdução básica de comunicação de rede	221
10.2.	Porta de Comunicação	223
10.3.	Comunicação do CLP	224
10.3.1.	Comunicação de porta livre	224
10.3.2.	Comunicação MODBUS.....	227
10.3.3.	Comunicação CAN.....	230
10.3.4.	Desempenho de rede.....	231
11.	Capítulo Adicional	232
11.1.	Como chamar o arquivo de ajuda	232
11.2.	Como forçar os pontos de entrada e saída	233
11.3.	Faixa de valor da analógica	235
11.4.	Endereço do módulo de expansão.....	235
11.5.	Formulário de endereço do módulo de expansão.....	236
11.6.	Define o endereço da extensão com um interruptor de discagem	236
11.7.	Intervalo do endereço de host do CLP	237
12.	Exemplos	238
12.1.	Exemplo instrução LCD.....	238
12.2.	Porta serial porta livre.....	247

12.3.	Comunicação MODBUS.....	250
12.4.	Exemplo da instrução PID	254
12.5.	PID example 2	257
12.6.	Exemplo de Bit Lógico	261
12.7.	Exemplo de instrução Clock	262
12.8.	Exemplo de instrução de comunicação	263
12.9.	Exemplo de comparação de número Real	264
12.10.	Exemplo de comparação de String.....	265

1. Introdução

Parabéns por seu Micro CLP fornecido pela Metaltex Ltda.

O Micro CLP Nexo é uma CPU compacta e expansível que substitui mini-CLPs, múltiplos temporizadores, relés e contadores.

O Micro CLP Nexo se encaixa perfeitamente no espaço entre os relés de temporização e os CLPs low-end. Cada CPU incorpora não apenas um relógio e calendário em tempo real, mas também fornece suporte para módulos de E / S de expansões opcionais para aprimorar aplicações de controle e monitoramento.

Os ajustes de dados podem ser facilmente realizados através do teclado, do visor LCD ou do programa NeXoLadder. As opções de trilho DIN e painel estão disponíveis, oferecendo flexibilidade total às várias necessidades de instalação de sua aplicação.

O Micro CLP Nexo está disponível em versões de 120V / 240V AC ou 12V e 24V DC, tornando-se a solução ideal para substituição de relés ou aplicações de controle simples como iluminação de prédios e estacionamento, gerenciamento de iluminação automática, controle de acesso, sistemas de rega, controle de bombas, Sistemas de ventilação, automação residencial e um amplo campo de outras aplicações exigentes de baixo custo.

Recomendamos a leitura deste manual antes de colocar o Micro CLP Nexo para funcionar. A instalação, programação e utilização da unidade são detalhadas neste manual. O Micro CLP Nexo é rico em recursos, oferece um modo de operação off-line, permitindo configuração completa e testes antes do comissionamento do serviço no campo. Ao rever este manual, você descobrirá muitas outras propriedades vantajosas do produto, o que simplificará e otimizará muito o uso do seu Micro CLP Nexo.

2. Validade deste manual

O manual se aplica aos dispositivos CLPs da linha NEX:

NEX14-DR
NEX16E-AR
NEX16E-DR
NEX18-AR
NEX18-DR
NEX18-DRT
NEX-EA4I
NEX-SA2VI

3. Instruções de Segurança

Este manual contém avisos que devem ser observados para garantir a sua segurança pessoal, bem como para evitar danos materiais. Os avisos referentes à sua segurança pessoal são destacados no manual por um símbolo de alerta de segurança; Avisos referentes a danos à propriedade apenas não têm símbolo de alerta de segurança. Os avisos abaixo são classificados de acordo com o grau de perigo.

Cuidado

Indica que podem ocorrer mortes ou ferimentos graves se não forem tomadas as devidas precauções.

Com um símbolo de alerta de segurança indica que podem ocorrer lesões pessoais menores se não forem tomadas as devidas precauções.

Cuidado

Sem um símbolo de alerta de segurança indica que podem ocorrer danos à propriedade se não forem tomadas as devidas precauções.

Atenção

Indicar que um resultado ou situação não intencional pode ocorrer se o aviso correspondente não for levado em consideração.

Se existir mais do que um grau de perigo, será utilizada a advertência que representa o maior grau de perigo. Um aviso de aviso de lesões a pessoas com um símbolo de alerta de segurança também pode incluir um aviso relativo a danos à propriedade.

4. Prescrições de Uso

Observe o seguinte:

Atenção

Este dispositivo e seus componentes só podem ser utilizados para as aplicações descritas no catálogo ou na descrição técnica e apenas em conexão com dispositivos ou componentes de outros fabricantes aprovados ou recomendados pela Metaltex.

O funcionamento correto e confiável do produto requer transporte, armazenamento, posicionamento e montagem adequados, bem como operação e manutenção cuidadosa.

Isenção de Responsabilidade

Analisamos o conteúdo desta publicação para garantir a consistência com o hardware e software descritos.

Como a variância não pode ser totalmente excluída, não podemos garantir total coerência.

No entanto, as informações nesta publicação são revistas regularmente e as correções necessárias são incluídas nas edições subsequentes.

Suporte Adicional

Estamos orgulhosos em responder sua pergunta assim que pudermos:

Por favor, consulte o nosso site em www.metaltex.com.br para seu ponto de contato mais próximo ou envie um e-mail para engenharia@metaltex.com.br

4.1. Instalação / remoção do CLP Nexo

Dimensões:

As dimensões de instalação do CLP Nexo são compatíveis com trilho DIN 43880.

O CLP Nexo pode ser encaixado de forma a ser montado em trilhos DIN de 35 mm de acordo com EN 50022 ou na parede.

CLP Nexo Largura:

- NEX14, NEX18 tem uma largura de 95mm.
- Os módulos de expansão têm uma largura de 72mm.

Atenção

Sempre desligue a alimentação antes de "remover" e "inserir" um módulo de expansão.

4.2. Montagem em trilho DIN

Montagem

Como montar um módulo CLP Nexo e um módulo de expansão em um trilho DIN:

1. Encaixe o módulo básico CLP Nexo no trilho.
2. Empurre a extremidade inferior para encaixá-la. O bloqueio de montagem na parte traseira deve encaixar.
3. Encaixe o módulo de expansão CLP Nexo no trilho.
4. Deslize o módulo para a esquerda até tocar na CPU do CLP.
5. Empurre a extremidade inferior para encaixá-la. O bloqueio de montagem na parte traseira deve encaixar.
6. Remova a tampa de plástico na porta de expansão da CPU e do módulo de expansão.

Além disso, o conector no cabo plano para a CPU.



Repita os passos do módulo de expansão para montar outros módulos de expansão.

Remoção

Remoção do CLP Nexo:

Se você instalou apenas um CLP Nexo básico:

1. Insira uma chave de fenda no olhal na parte inferior do bloqueio de deslizamento e mova a trava para baixo.
2. Desloque o CLP Nexo básico do trilho DIN.

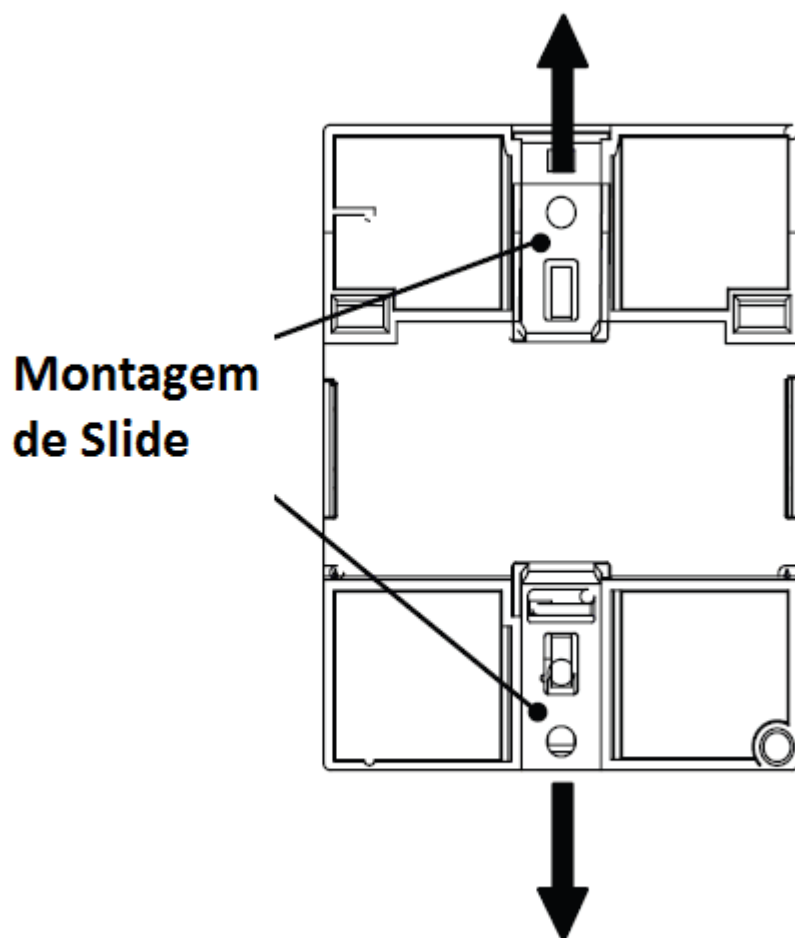
Se você tiver conectado pelo menos um módulo de expansão ao CLP Nexo básico:

1. Remova o conector do cabo.
2. Deslize o módulo de expansão para a direita.
3. Insira uma chave de fenda no olhal na parte inferior do bloqueio da corrediça e mova a alavanca para baixo.
4. Desloque o módulo de expansão do perfil.

Repita os passos 1 ao 4 para todos os outros módulos de expansão.

4.3. Montagem com Parafuso

Para montagem com parafuso, deslize primeiro as guias de montagem na parte traseira dos dispositivos para o exterior. Agora você pode montar o CLP Nexo na parede por meio de duas lâminas de montagem e dois parafusos $\varnothing M4$ (torque de aperto de 0,8 a 1,2 Nm).



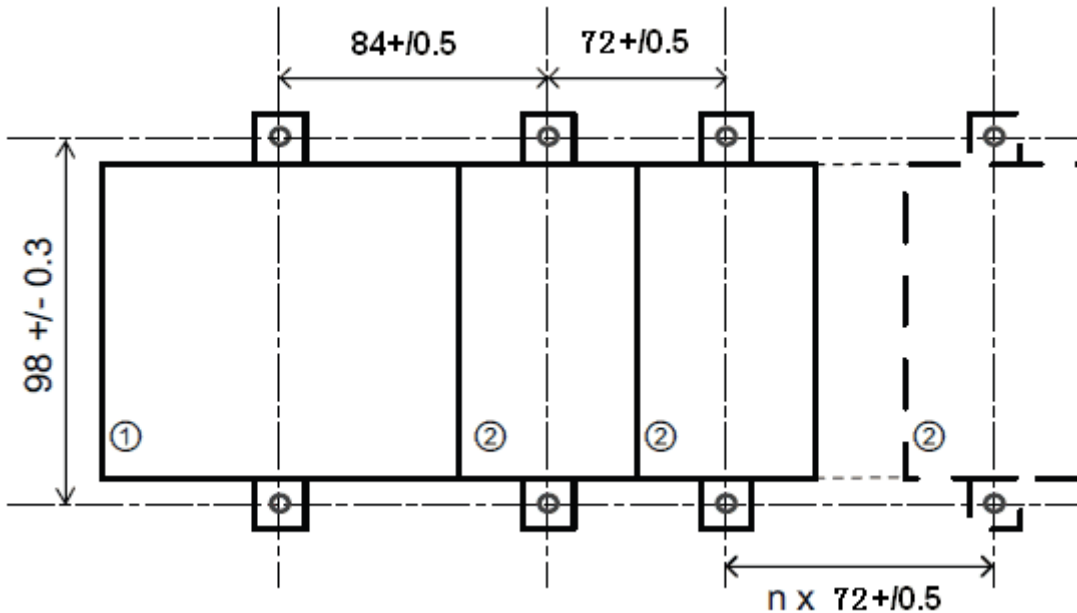
Molde de perfuração para montagem com parafuso

Antes de montar o CLP Nexo com parafuso, você precisa perfurar usando o modelo mostrado abaixo. Todas as dimensões em mm

Orifício de furação para parafuso $\varnothing M4$, torque de aperto de 0,8 a 1,2 Nm

CPU Linha NEX

Linha NEX14 e NEX18:



1. NEX18 CPU. NEX-E extensão

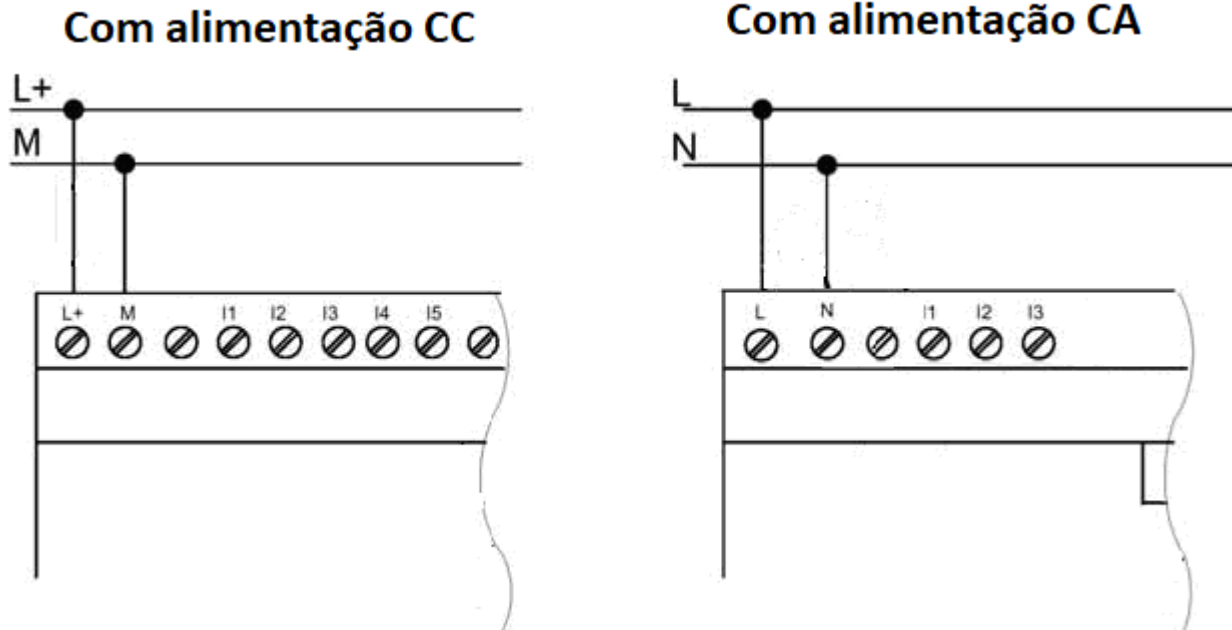
4.4. Conexão da Fonte de Alimentação

As versões NEX18-AR e NEX14-AR, são adequadas para operação com tensões nominais de 110 V CA e 240 VCA. As versões NEX18-DRT, NEX18-DR e NEX14-DR podem ser operadas com uma fonte de alimentação de 12 ou 24 VDC.

Nota

Uma falha de energia pode causar um sinal adicional.
Dados do último ciclo ininterrupto são armazenados no Nexo.

Para conectar o CLP Nexo a fonte de alimentação:



4.5. Conectando as entradas

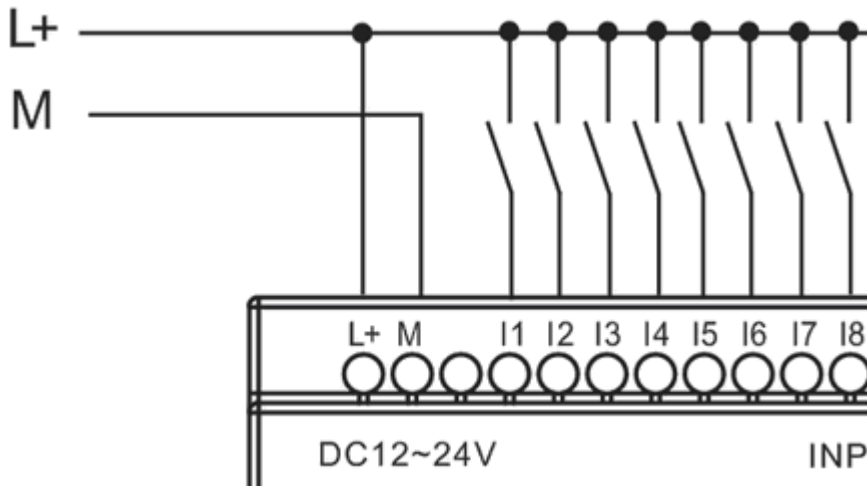
Requisitos

Nas entradas você conectar elementos sensores, tais como: barreiras de luz, interruptores momentâneos, comutadores, interruptores de controle de luz, etc.

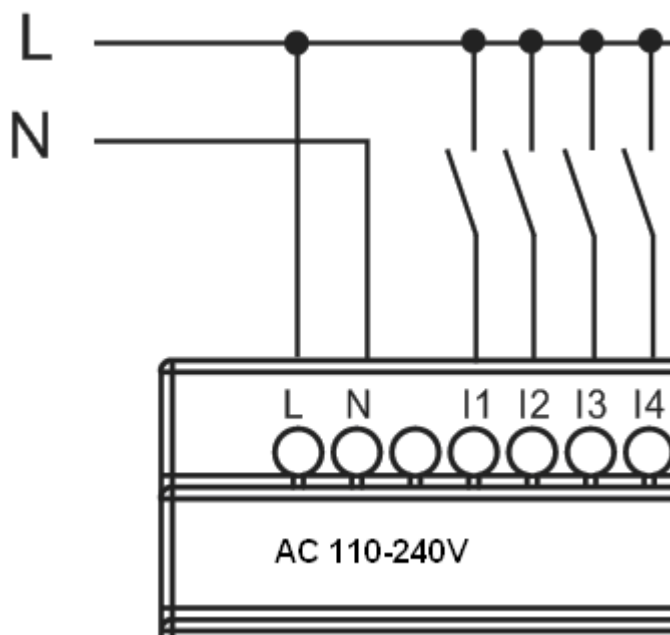
	Tipo AC	Tipo DC
Sinal status 0 Corrente de Entrada	<40VAC <0.03mA	<5VDC <0.1mA
Sinal status 1 Corrente de Entrada	>79VAC de 0.06 0.24mA	>10VDC de 0.3mA
Entrada Analógica	Não se aplica	EA0-EA6(0-10V DC)

Ligação das entradas são mostradas nas figuras abaixo:

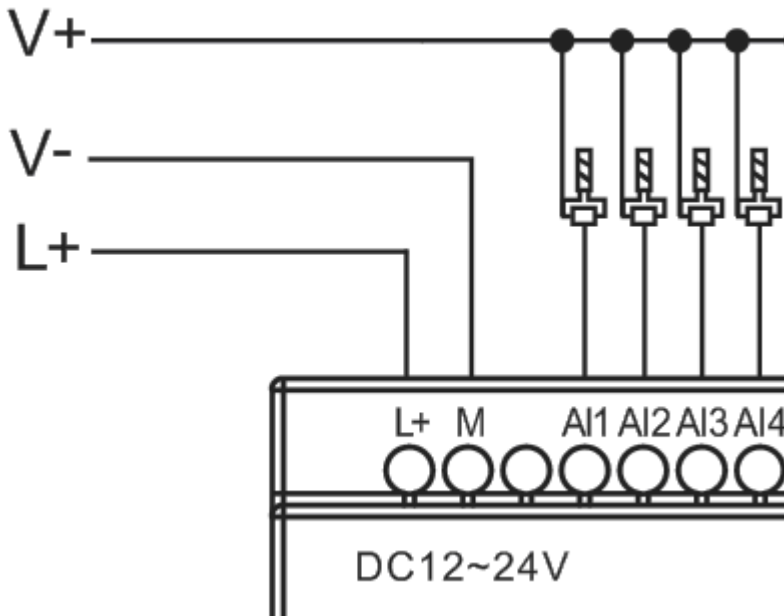
Entradas digitais CC



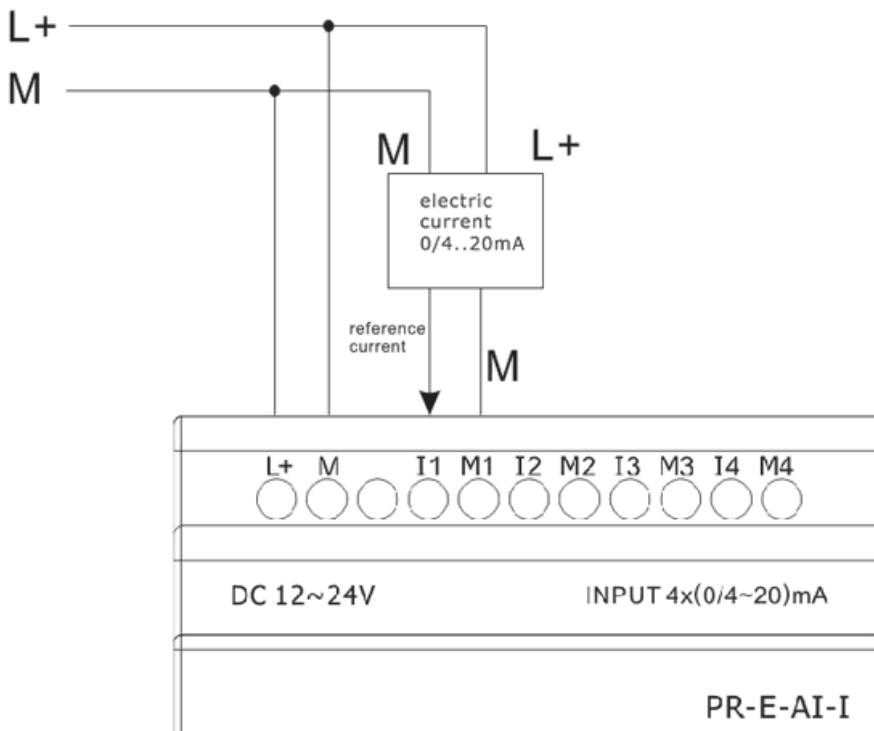
Entradas digitais CA



Entradas Analógicas de tensão (0...10VCC)



Entradas Analógicas de corrente (0...20mA)

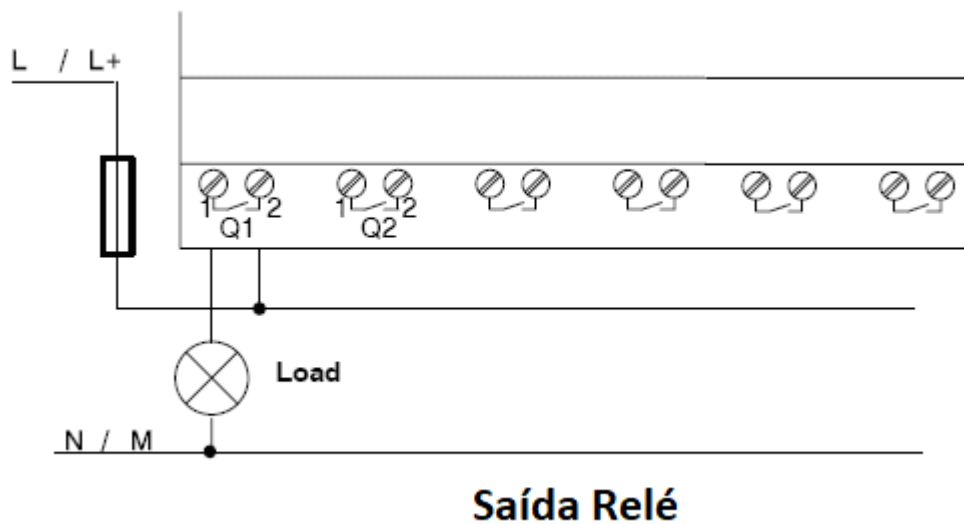


A figura acima mostra como fazer uma medição de corrente de quatro fios.

4.6. Conectando as Saídas

Requisitos para saída relé

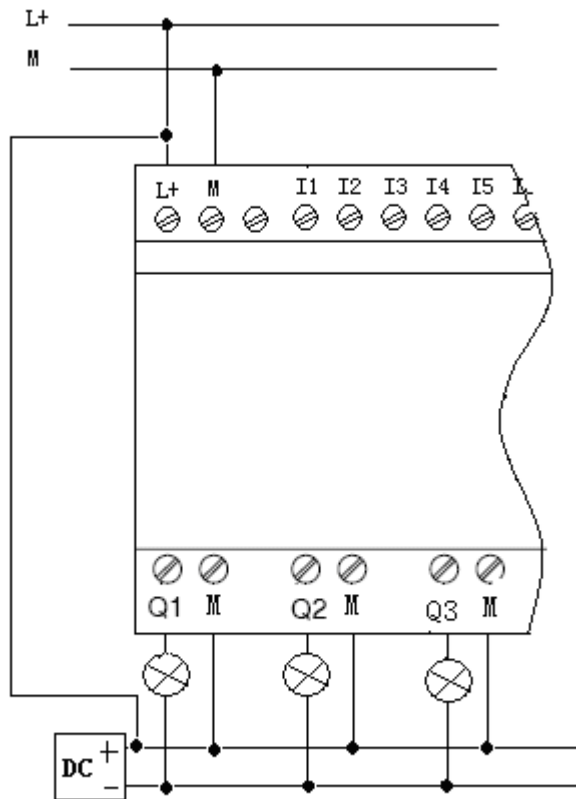
Várias cargas tais como, lâmpada fluorescente, motor, contatores, etc., podem ser conectadas às saídas do Nexo. O valor máximo de corrente nos contatos de saída é de 10A para a cargas resistivas e 3A para cargas indutivas. A conexão está de acordo com a figura a seguir:



Requisitos para saídas a transistor:

A carga deve ter as seguintes características:

- A máxima corrente chaveada não pode exceder 0.3A.
- Quando a saída está em ON (Q=1), a máxima corrente é de 0.3A.

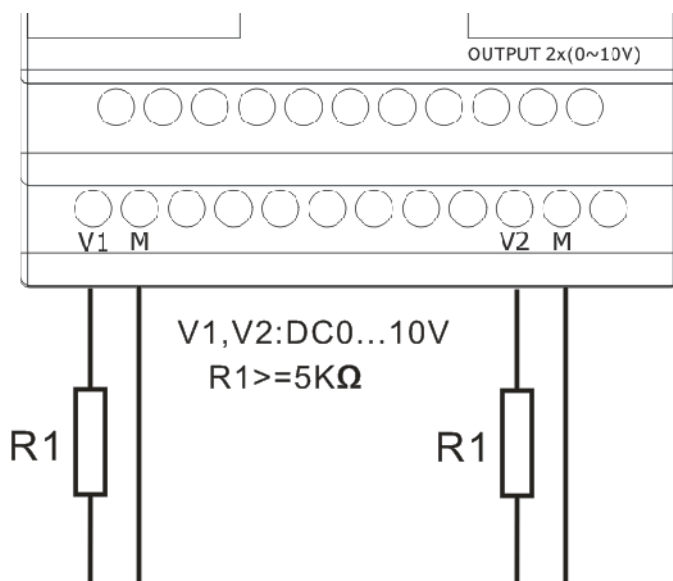


Saída Transistor PNP

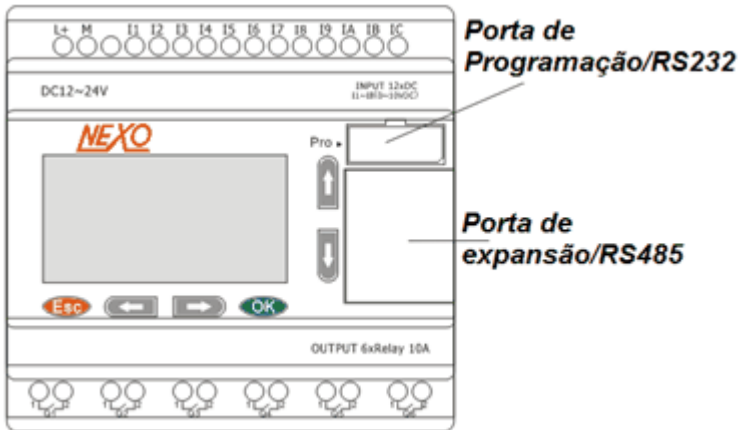
Notas (PNP):

- A tensão da carga deve ser $\leq 60V_{CC}$ (deve ser CC).

NEX-SA2VI (Saída analógica de corrente ou tensão).

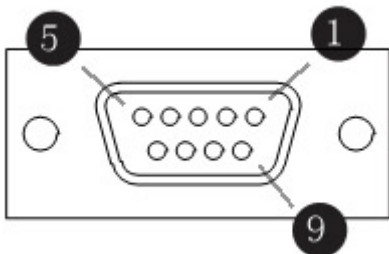


4.7. Instruções da porta de comunicação



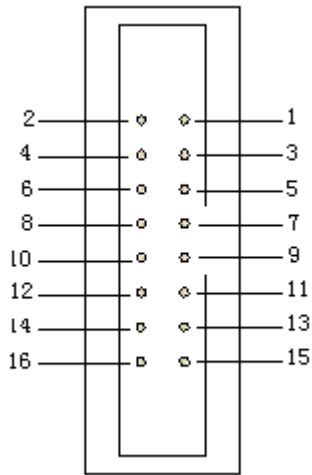
1. Porta de programação/RS232 (Cabo RS232, cabo USB, cabo RS485).

Se a porta de programação for usada como RS232 comum segue a pinagem do cabo NEX-C-RS232:



PINOS	FUNÇÕES
2	RXD
3	TXD
5	
Outros	NULL

1. Porta de expansão/RS485 (pinagem)



- 3-----RS485 A
- 5-----RS485 B
- 4-----GND
- 6-----GND
- 7-----CANL
- 9-----CANH
- 15-----+5V
- 16-----+5V

5. Guia de referência

5.1. Área de memórias especiais

SMB0

Always_On	SM0.0	Bit sempre em 1
First_Scan_On	SM0.1	Bit em 1 apenas no primeiro escan
Clock_60s	SM0.4	30 segundos DESL, 30 segundos LIG
Clock_1s	SM0.5	0.5 segundos DESL, 0.5 segundos LIG

SMB1

Result_0	SM1.0	Bit ligado quando certas operações executadas retornam 0(zero) result = 0
Overflow_Illegal	SM1.1	
Neg_Result	SM1.2	Bit ligado quando uma operação matemática produz um número negativo
Divide_By_0	SM1.3	Bit ligado quando há tentativa de divisão por 0(zero)
Table_Overflow	SM1.4	Bit ligado quando a tabela de instrução está cheia
Table_Empty	SM1.5	Bit ligado quando uma LIFO ou FIFO recebe uma tabela vazia
Not_BCD	SM1.6	Bit ligado quando tenta converter um valor não BCD para um valor binário
Not_Hex	SM1.7	Bit ligado quando um valor ASCII não pode ser convertido para um valor hexadecimal válido

Endereço interno das variáveis dos botoes do LCD:

ESC → SM190.0
 OK → SM190.1
 UP → SM190.2
 DOWN → SM190.3
 LEFT → SM190.4
 RIGHT → SM190.5

Quando o valor de SM192.0 for 1, o LCD estará ligado.
 Quando o valor de SM192.0 for 0, o LCD estará apagado.

SMW22-SMW26 Tempo de Escaneamento

SMW22 Tempo de scan do último scan.
 SMW24 Tempo de scan mínimo.
 SMW26 Tempo de scan máximo.

Eventos de interrupção

I1.2 Rising edge	PLC_EVENT_INPUTP0	0	Maior Prioridade
I1.4 Rising edge	PLC_EVENT_INPUTP1	1	Alta Prioridade
Timer interrupt 0	PLC_EVENT_TIMER0	10	Baixa Prioridade
Timer interrupt 1	PLC_EVENT_TIMER1	11	Menor Prioridade

NEX14-DR, NEX18-DR e NEX18-DRT

I1.0	HC0
I1.1	
I1.2	HC2
I1.3	HC3

Cada contador de alta velocidade ocupa uma entrada para receber o pulso.

Modelos suportados

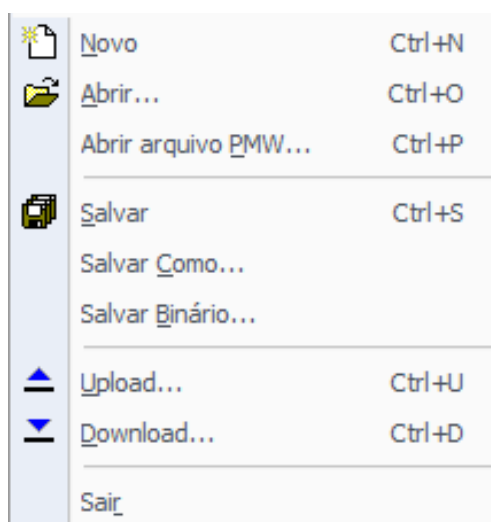
NEX14-AR
NEX14-DR
NEX18-AR
NEX18-DR
NEX18-DRT

6. NeXo Ladder informações de uso

6.1. Menu principal

Arquivo Editar Vista PLC Depurar Ajuda

6.1.1. Arquivo



Novo: Usado para criar um novo projeto, assim como o atalho CTRL + N.

Abrir: Abrir um projeto (*.VCW), também pode ser aberto com CTRL + O. Se precisar abrir um arquivo de PWM (*.PWM) você pode usar o atalho CTRL + P.

Salvar: Salvar o projeto em edição, atalho CTRL + S.

Salvar como: Você pode salvar o projeto com outro nome.






Salvar Binário: Salvar o projeto no formato binário

Upload: Carregar o programa do CLP para o PC, assegure que a comunicação esteja OK. O programa será carregado para um novo projeto, poderá ser salvo e nomeado. Use o atalho CTRL + U se preferir.

Download: Enviar o programa para o CLP, compile o seu programa, você também poderá enviá-lo ao CLP usando CTRL + D.

Sair: Fechar o NeXo Ladder®.

6.1.2. Editar

	Desfazer	Ctrl+Z
	Refazer	Ctrl+Y
<hr/>		
	Recortar	Ctrl+X
	Copiar	Ctrl+C
	Colar	Ctrl+V
	Selecionar Tudo	Ctrl+A
<hr/>		
	Encontrar e Substituir	Ctrl+F

Desfazer: Desfazer a última ação, você pode usar esse comando múltiplas vezes. Se você abrir, salvar, ou compilar o projeto o buffer será zerado. É possível usar esse comando também pelo atalho CTRL + Z.

Refazer: Refazer a última ação.

Recortar: Remove objeto(s) selecionado(s) e armazena na clipboard do sistema.

Copiar: Copia objeto (s) selecionado (s) para clipboard do sistema.

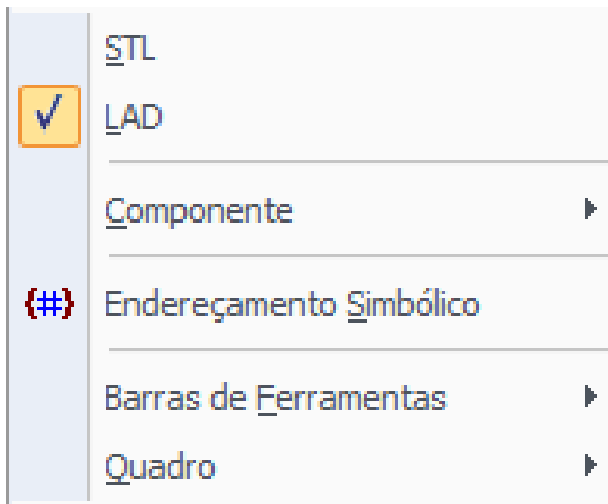
Colar: Cola o objeto que está no clipboard.

Selecionar Tudo: Seleciona tudo, também feito através do atalho CTRL+A.

Encotrar e Substituir: Usado para “procurar”, “substituir” e “ir para” textos no programa, na tabela de variáveis, bloco de dados, tabela de símbolos e tabela de estado, você também pode usar o atalho CTRL+F.

Sair: Fechar o NeXo Ladder®.

6.1.3. Exibir



STL: Mostra o programa em instruções STL (Lista de Texto).

LAD: Mostra o programa em LAD (Ladder).

Componente: Visualizar alguns componentes como, bloco de dados, blocos de sistema, programa em edição, símbolos de funções, variáveis, referência cruzada, configuração de comunicação.

Endereçamento Simbólico: Mostra os símbolos das variáveis.

Program Editor Variable Symbol x

	Symbol	Adress	Data Type	Comment
<input checked="" type="checkbox"/>	Start	M0.0	BOOL	
<input checked="" type="checkbox"/>	Stop	M0.2	BOOL	
<input checked="" type="checkbox"/>	Motor	M0.3	BOOL	
<input type="checkbox"/>			BOOL	

Barra de Ferramentas: Mostra ou oculta barra de ferramentas.

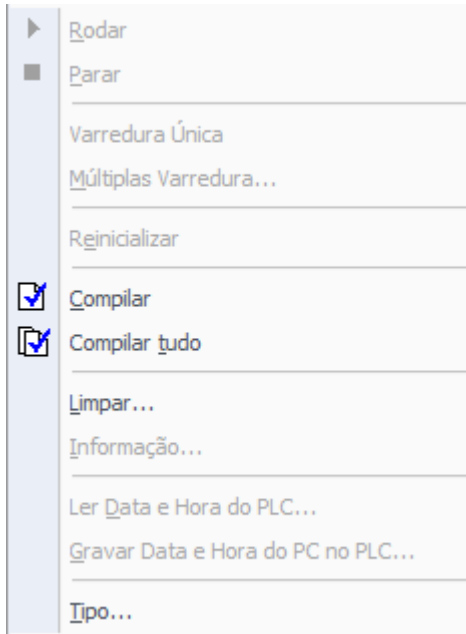
- Padrão
- Depurar
- Instrução
- Instrução (Modo Fx)

Você pode escolher ferramentas á mostrar, por padrão são todas ativas.

Quadro: Janelas de operação. Mostrar ou esconder:

- Status Chart
- Project Manager
- Information Output
- Debug Information

6.1.4. CLP



Rodar: Mudar CLP para o modo Rodar, requer comunicação entre o CLP e o programa.

Parar: Muda o CLP para STOP.

Compilar: Compilar o programa da página atual.

Compilar tudo: Compilar o projeto todo (Programa, Dados e Sistema).

Limpar: Limpar todos os dados do CLP, permitido somente Off-line.

Informação: Ler informações do CLP.

Gravar Data e Hora do PC no CLP: Escrever data/hora do CLP, requer comunicação ok.

Tipo: Escolher tipo de CLP.

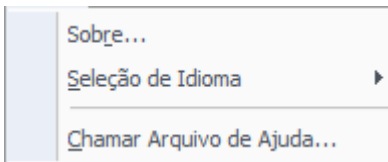
6.1.5. Depurar



Conectar: Mostra dados do CLP na janela de edição (monitoramento).

Desconectar: No Desliga o monitoramento. Isso não coloca o CLP em modo STOP, para colocar em STOP use o menu PLC > Stop.

6.1.6. Ajuda



Sobre: Informações sobre o programa.

Seleção de Idioma: Selecciona o idioma, Português ou Inglês.

Chamar Arquivo de Ajuda: Abre o arquivo de ajuda.

6.2. Barra de Ferramentas



Novo: Usado para criar um novo projeto, assim como o atalho CTRL + N.

Abrir: Abre um projeto (*.VCW), também pode ser aberto com CTRL + O. Se precisar abrir um arquivo de PWM (*.PWM) você pode usar o atalho CTRL + P.

Salvar: Salva o projeto em edição, atalho CTRL + S.

Desfazer: Desfaz a última ação, você pode usar esse comando múltiplas vezes. Se você abrir, salvar, ou compilar o projeto o buffer será zerado. É possível usar esse comando também pelo atalho CTRL + Z.

Refazer: Refaz o que foi desfeito.

Recortar: Remove objeto(s) selecionado(s) e armazena na clipboard do sistema.

Copiar: Copia objeto(s) selecionado(s) para clipboard do sistema.

Colar: Cola o objeto que está na clipboard.

Endereçamento simbólico: Mostra os símbolos das variáveis.

Compilar: Compila o programa da página atual.

Compilar tudo: Compila o projeto todo (Programa, Dados e Sistema).

Upload: Carrega o programa do CLP para o PC. O programa será carregado para um novo projeto, poderá ser salvo e nomeado.

Download: Compila o programa e envia para o CLP.

Conectar: Mostra dados do CLP na janela de edição (monitoramento).

Disconectar: Desliga o monitoramento.

Rodar: Muda CLP para modo RUN, requer comunicação ok do CLP com o software.

Parar: Para as instruções do menu principal.

Apagar: Apaga a instrução selecionada.

Selecionar: Seleciona ferramenta.

Contato normalmente aberto: Insere contato aberto.

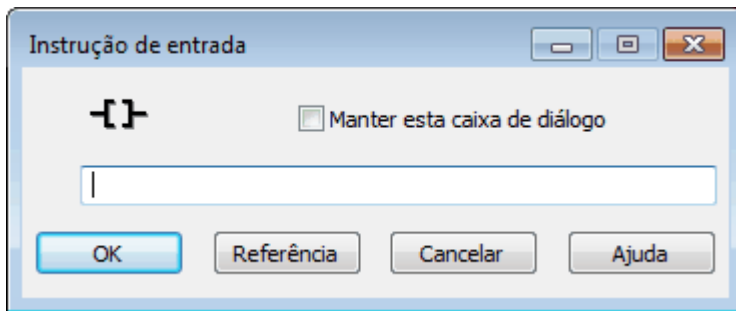
Contato normalmente fechado: Insere contato fechado.

Borda de subida: Insere borda de subida do contato.

Borda de descida: Insere borda de descida do contato.

Bobina: Insere bobina. Deve ser inserida no fim da linha.

Instrução de entrada: Insere funções especiais. Na janela como mostrado:



Entre com a instrução na janela, ou use o botão "Referência".

Nível: Conecta funções, contato e bobina com uma linha horizontal.

Linha horizontal: Insere uma linha horizontal.

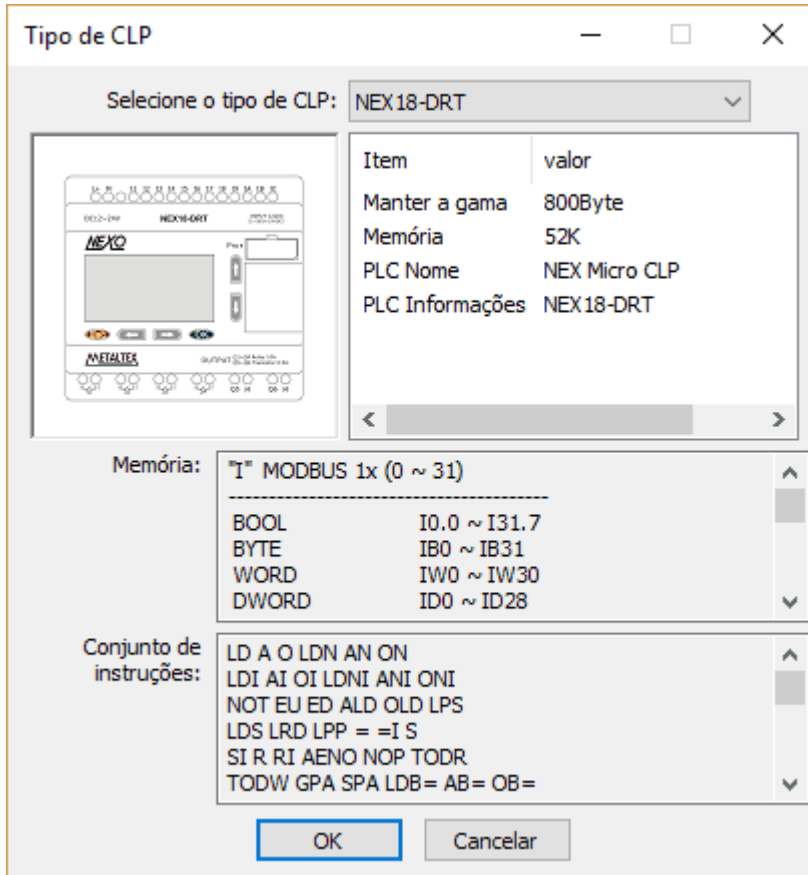
Linha vertical: Insere uma linha vertical.

Inverter linha: Inverter da linha da linha.

6.3. Instruções

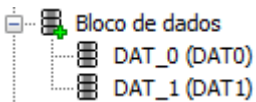


6.3.1. Projeto



Para selecionar o tipo de CLP, quando selecionado, os parâmetros aparecerão na árvore de instruções.

6.3.2. Bloco de dados



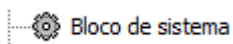
No Bloco de dados temos DAT-0 e DAT-1, você pode inserir um Data Block novo com o botão direito do mouse.

O bloco de dados tem a seguinte estrutura:

	endereço	Tipo de dados	valor	comentário
		BOOL		
		BOOL		
		BOOL		
		BOOL		

No bloco de dados, você pode inserir o endereço, tipo, valor e comentário de uma variável. O conteúdo do data block é escrito no CLP depois do download, e permanece até que um programa novo seja enviado ao CLP.

6.3.3. Bloco de Sistema



Dê um duplo click em Bloco de sistema para abrir a janela:

Bloco do Sistema ✕

Memória Retentiva	Interrupção	Forçar Saídas	
RS232/RS485	RS232/RS485	CAN	Senha

	Porta 0	Porta 1	
Protocolo:	<input type="text" value="Modbus"/>	<input type="text" value="Modbus"/>	
Estação:	<input type="text" value="1"/>	<input type="text" value="1"/>	(Intervalo 0... 255)
Velocidade:	<input type="text" value="9600 bps"/>	<input type="text" value="9600 bps"/>	
Bit dados:	<input type="text" value="8 (RTU)"/>	<input type="text" value="8 (RTU)"/>	
Paridade:	<input type="text" value="NONE"/>	<input type="text" value="NONE"/>	
b.Parada:	<input type="text" value="1 Bit"/>	<input type="text" value="1 Bit"/>	
Tempo Resposta (100ms)	<input type="text" value="10"/>	<input type="text" value="10"/>	(Intervalo 1... 255)
Intervalo de quadros (B)	<input type="text" value="10"/>	<input type="text" value="10"/>	(Intervalo 1... 255)

As modificações só terão efeito após o Download

Interface RS232/RS485: Todas as portas usam o protocolo MODBUS. Você pode usar até 4 portas: porta 0, porta 1, porta 2 e porta 3.

Interface CAN:

The screenshot shows the 'Bloco do Sistema' configuration window. At the top, there are three tabs: 'Memória Retentiva' (set to RS232/RS485), 'Interrupção' (set to RS232/RS485), and 'Forçar Saídas' (set to Senha). Below the tabs, there is a 'Padrão' button. The main area is divided into two columns for 'Porta 0' and 'Porta 1'. For each port, the 'Taxa' is set to 500,000 bps, 'Estação' is 1, 'Rx Msg ID' is 22, and 'Tx Msg ID' is 21. The 'Tempo reset' is set to 500 ms, with a note '(10 - 1000ms)'. At the bottom, there is a note 'As modificações só terão efeito após o Download' and 'OK' and 'Cancelar' buttons.

O Nexo suporta comunicação CAN. Podemos aprender sobre ela no Bloco de comunicação.

Interface de Senhas:

The screenshot shows the 'Bloco do Sistema' configuration window. At the top, there are three tabs: 'Memória Retentiva' (set to RS232/RS485), 'Interrupção' (set to RS232/RS485), and 'Forçar Saídas' (set to CAN). Below the tabs, there is a 'Padrão' button. The main area is titled 'Segurança' and contains three radio button options: 'Nível 1: Livre' (selected), 'Nível 3: Mínimo', and 'Nível 4: Total'. Below these are two text input fields labeled 'SENHA' and 'Confirma'. At the bottom, there is a note 'As modificações só terão efeito após o Download' and 'OK' and 'Cancelar' buttons.

Temos 3 níveis de senhas.

Level 1 - Fraca: todas as funções, sem restrições.

Level 2 - Mínima: necessário senha antes de usar qualquer função.

Level 3 - Forte: Upload e Download bloqueado e necessário senha antes de usar qualquer função.

Interface Áreas Retentivas:

	Área	Inicia em	Tamanho	
Bloco 0	VB	0	256	Limpar
Bloco 1	VB	0	0	Limpar
Bloco 2	T	0	32	Limpar
Bloco 3	T	64	32	Limpar
Bloco 4	C	0	16	Limpar
Bloco 5	MB	14	18	Limpar

Limpar Memória Limpar Retentiva

As modificações só terão efeito após o Download

OK Cancelar

Por padrão, toda área de memória M, T, V, e C são configuradas como retentivas. É possível redefinir essas áreas para não retentivas. Para os temporizadores e contadores, somente a memória do valor atual poderá ser mantida. Os bits são resetados quando o Nexo é desligado.

Todas as variáveis armazenadas na área retentiva são gravadas permanentemente. O Nexo pode armazenar até 800 bytes.

Configuração de Interrupção:

Bloco do Sistema

RS232/RS485 RS232/RS485 CAN Senha

Memória Retentiva Interrupção Forçar Saídas

Padrão

Tempo Interrup 0 (1ms) 200 (Intervalo 1 ... 255)

Tempo Interrup 1 (1ms) 200 (Intervalo 1 ... 255)

As modificações só terão efeito após o Download

OK Cancelar

O Nexo possui 2 eventos de interrupção por tempo, interrupção 0 e interrupção 1. Você pode ajustar o tempo de interrupção entre 1 e 255 milissegundos.

Tabela Forçar Saídas:

Bloco do Sistema

RS232/RS485 RS232/RS485 CAN Senha

Memória Retentiva Interrupção Forçar Saídas

Padrão

Saídas

	0	1	2	3	4	5	6	7
Q0.x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q1.x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q2.x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q3.x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Saídas

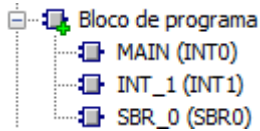
	0	1	2	3	4	5	6	7
Q4.x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q5.x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q6.x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Q7.x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

As modificações só terão efeito após o Download

OK Cancelar

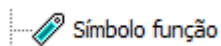
As saídas podem ser forçadas para 1, o estado forçado se mantém até mesmo quando o Nexo está em modo STOP.

6.3.4. Bloco de Programa



O bloco de programa possui três partes, nomeadas de MAIN (Programa principal), INT-1 (rotina de interrupção) e SBR-0 (sub-rotina). Selecione qualquer um deles e clique com o direito para remover ou adicionar um novo INT ou SBR. O MAIN não pode ser removido.

6.3.5. Símbolo de função



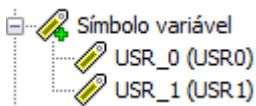
De um duplo clique em "Símbolo função" para abrir a interface:

		símbolo	endereço	comentário
	✓	MAIN	INT0	
	✓	INT_1	INT1	
	✓	SBR_0	SBR0	

Você pode mudar os símbolos, endereços e comentários.

		símbolo	endereço	comentário
	✓	Mk7	INT0	Variável de contagem
	✓	MTX	INT1	Principal
	✓	ALN	SBR0	Teste

6.3.6. Símbolo de variável



De um duplo clique para abrir:

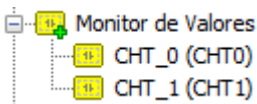
		símbolo	endereço	Tipo de da...	comentário
				BOOL	
				BOOL	
				BOOL	
				BOOL	

É possível alterar os símbolos, endereços, tipos e comentários:

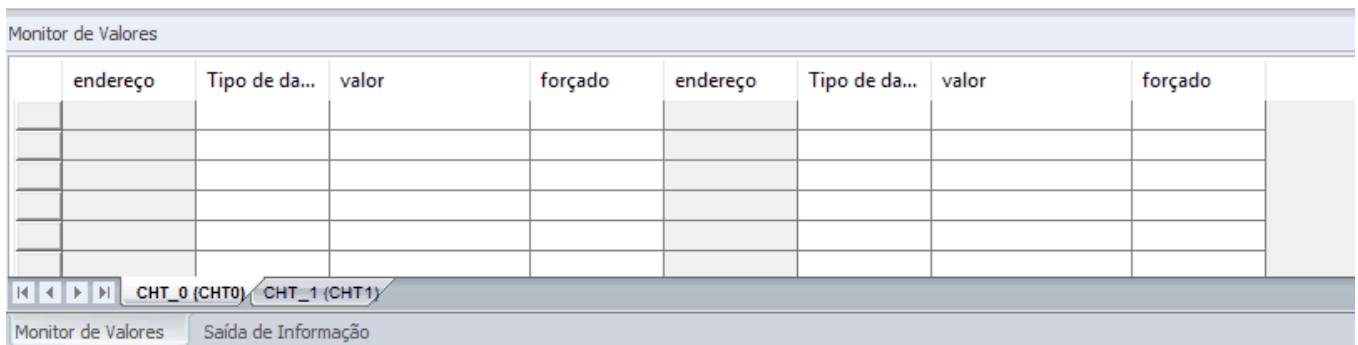
	símbolo	endereço	Tipo de da...	comentário
✓	Iniciar	I0.0	BOOL	Início do processo
✓	Parar	I0.1	BOOL	Parar processo
	Alarme	Q0.0	BYTE	Alarme
			BOOL	

Quando o endereço e o tipo não coincidem o endereço fica vermelho.

6.3.7. Monitor de valores

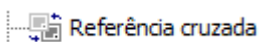


Duplo clique para abrir:



No Monitor de valores você pode monitorar e forçar valores.

6.3.8. Referência cruzada

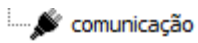


A referência cruzada mostra o endereço, símbolo, localização e instrução como na imagem abaixo:

Símbolo de Função		Símbolo variável		Editor de Programas		Bloco de dados		Referência Cruzada	
endereço	símbolo	localização	contexto						
I0.0	Iniciar	ALN (SBR0) NETWORK 1 Col 0 Row 0	- -						
I0.1	Parar	ALN (SBR0) NETWORK 1 Col 6 Row 0	-()						

Não é preciso fazer uma compilação para usar a referência cruzada.

6.3.9. Comunicação



Configuração de interface de comunicação com o Nexo:

comunicação

Porta serial Modbus TCP/IP CAN Kvaser CANpro CANalyst-II

Padrão

Estação: 0

Porta:

Parâmetros do barramento

Taxa de 9600 bps

Paridade: NONE

Bit de 1 Bit

OK Cancelar

Porta Serial: Configuração serial padrão.

MODBUS TCP/IP: Ainda não é suportado.

CAN Kvaser, CANpro, CANalyst-II são modos de comunicação CAN.

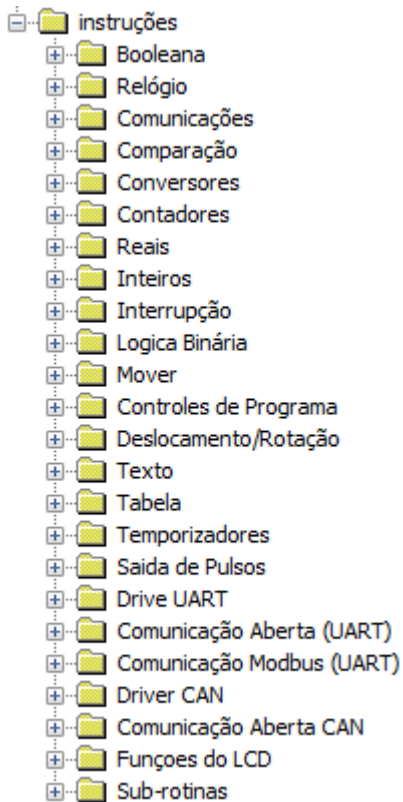
MODBUS TCP/IP: Ainda não suportado.

The screenshot shows a dialog box titled 'comunicação' with a close button (X) in the top right corner. It has five tabs: 'Porta serial', 'Modbus TCP/IP', 'CAN Kvaser', 'CANpro', and 'CANalyst-II'. The 'Modbus TCP/IP' tab is selected. Inside the dialog, there is a 'Padrão' button. Below it, there are four input fields: 'Estação:' with a dropdown menu showing '0', 'Adaptador:' with a dropdown menu showing '0.0.0.0 (Qualcomm Atheros AR8172/8176/8178 PC)', 'Endereço IP:' with a text box containing '192.168.0.191', and 'Porta:' with a text box containing '8888'. At the bottom, there are 'OK' and 'Cancelar' buttons.

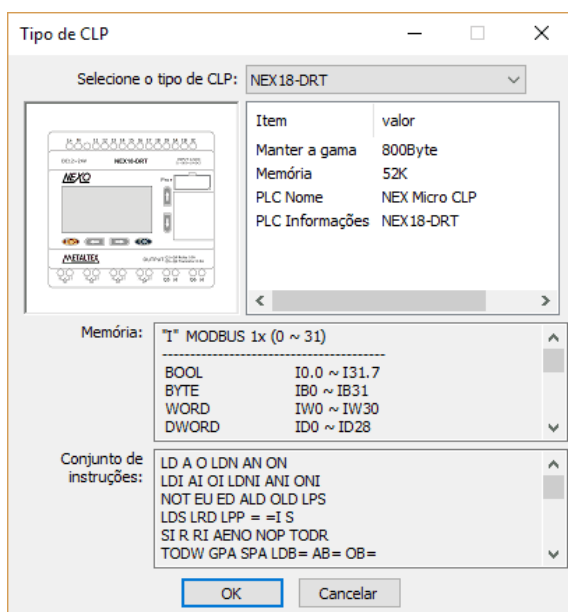
CAN Kvaser, CANpro, CANalyst-II são modos de comunicação CAN.

The screenshot shows the same 'comunicação' dialog box, but with the 'CAN Kvaser' tab selected. It features a 'Padrão' button, an 'Estação:' dropdown menu showing '0', and a 'Canal CAN:' dropdown menu. Below these are two checkboxes: 'Exclusivo' and 'Sem Acesso Inicial', both of which are unchecked. A section titled 'Parâmetros do barramento' contains three input fields: 'Velocidade:' with a dropdown menu showing '500,000 bps', 'Amostragem:' with a text box containing '50' and a '%' symbol, and 'SJW:' with a text box containing '1'. At the bottom, there are two text boxes for 'CAN Envelope: Rx' (containing '21') and 'Tx' (containing '22'). 'OK' and 'Cancelar' buttons are at the bottom right.

6.4. Instruções

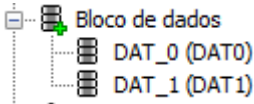


6.4.1. Projeto



As instruções serão explicadas detalhadamente na seção de instruções.

6.4.2. Bloco de dados



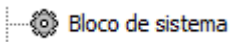
No Bloco de dados temos DAT-0 e DAT-1, você pode inserir um Data Block novo com o botão direito do mouse.

O bloco de dados tem a seguinte estrutura:

endereço	Tipo de dados	valor	comentário
	BOOL		
	BOOL		
	BOOL		
	BOOL		

No bloco de dados, você pode inserir o endereço, tipo, valor e comentário de uma variável. O conteúdo do data block é escrito no CLP depois do download, e permanece até que um programa novo seja enviado ao CLP.

6.4.3. Bloco de Sistema



De um duplo click em Bloco de sistema para abrir a janela:

Bloco do Sistema [X]

Memória Retentiva	Interrupção	Forçar Saídas
RS232/RS485	RS232/RS485	CAN Senha

[Padrão]

Porta 0	Porta 1	
Protocolo: Modbus	Modbus	
Estação: 1	1	(Intervalo 0... 255)
Velocidade: 9600 bps	9600 bps	
Bit dados: 8 (RTU)	8 (RTU)	
Paridade: NONE	NONE	
b.Parada: 1 Bit	1 Bit	
Tempo Resposta (100ms): 10	10	(Intervalo 1... 255)
Intervalo de quadros (B): 10	10	(Intervalo 1... 255)

As modificações só terão efeito após o Download

Interface RS232/RS485: Todas as portas usam o protocolo MODBUS. Você pode usar até 4 portas: porta 0, porta 1, porta 2 e porta 3.

É possível ajustar o baud rate, data bit, stop bit, paridade, timeout e o frame interval time (janela de varredura).

Interface CAN:

Memória Retentiva	Interrupção	Forçar Saídas
RS232/RS485	RS232/RS485	CAN Senha
Padrão		
	Porta 0	Porta 1
Taxa:	500,000 bps	500,000 bps
Estação:	1	1
Rx Msg ID:	22	22
Tx Msg ID:	21	21
Tempo reset:	500	500 (10 - 1000ms)
As modificações só terão efeito após o Download		
	OK	Cancelar

O Nexo suporta comunicação CAN. Podemos aprender sobre ela no Bloco de comunicação.

Interface de Senhas:

The screenshot shows a software configuration window titled "Bloco do Sistema". At the top, there are three tabs: "Memória Retentiva", "Interrupção", and "Forçar Saídas". The "Forçar Saídas" tab is active, showing a sub-tab for "Senha". Below this, there is a "Padrão" button. The main area is titled "Segurança" and contains three radio button options: "Nível 1: Livre" (selected), "Nível 3: Mínimo", and "Nível 4: Total". Below these are two text input fields labeled "SENHA" and "Confirma". A note states "FLivre: Todas as funções sem restrição." At the bottom of the window, a message reads "As modificações só terão efeito após o Download". The window has "OK" and "Cancelar" buttons at the bottom right.

Temos 3 níveis de senhas.

Level 1 - Fraca: todas as funções, sem restrições.

Level 2 - Mínima: necessário senha antes de usar qualquer função.

Level 3 - Forte: Upload e Download bloqueado e necessário senha antes de usar qualquer função.

Interface Áreas Retentivas:

Bloco	Área	Inicia em	Tamanho	Ação
Bloco 0	VB	0	256	Limpar
Bloco 1	VB	0	0	Limpar
Bloco 2	T	0	32	Limpar
Bloco 3	T	64	32	Limpar
Bloco 4	C	0	16	Limpar
Bloco 5	MB	14	18	Limpar

Limpar Memória Limpar Retentiva
 As modificações só terão efeito após o Download

Por padrão, toda área de memória M, T, V, e C são configuradas como retentivas. É possível redefinir essas áreas para não retentivas. Para os temporizadores e contadores, somente a memória do valor atual poderá ser mantida. Os bits são resetados quando o Nexo é desligado.

Todas as variáveis armazenadas na área retentiva são gravadas permanentemente. O Nexo pode armazenar até 800 bytes.

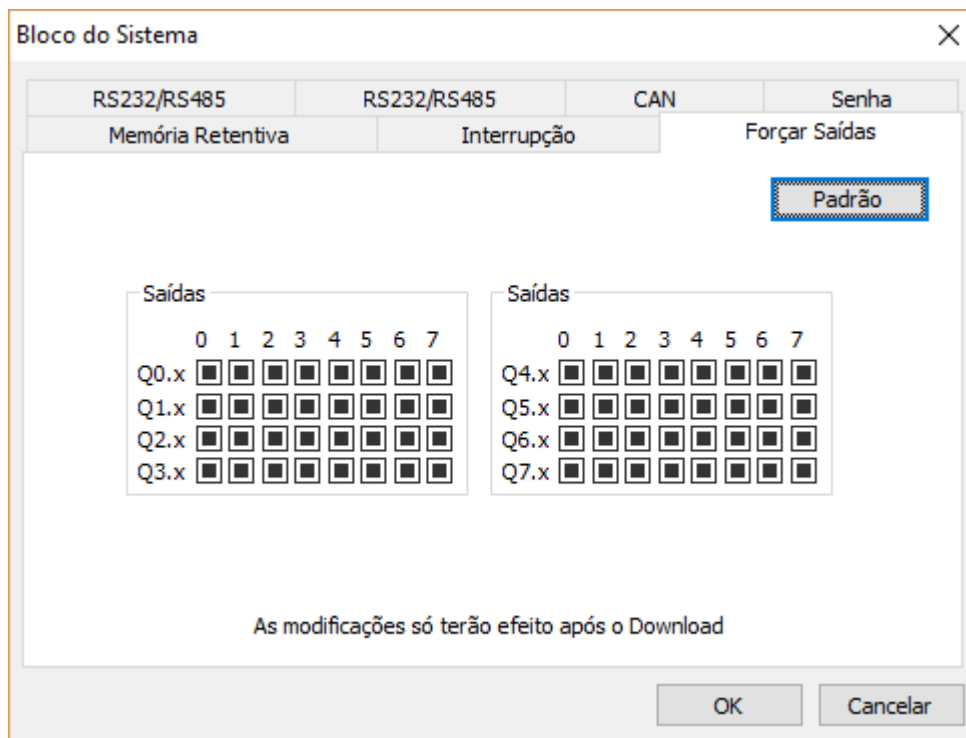
Configuração de Interrupção:

Tempo Interrup 0 (1ms) (Intervalo 1 ... 255)
 Tempo Interrup 1 (1ms) (Intervalo 1 ... 255)

As modificações só terão efeito após o Download

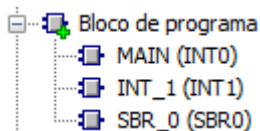
O Nexo possui 2 eventos de interrupção por tempo, interrupção 0 e interrupção 1. Você pode ajustar o tempo de interrupção entre 1 e 255 milissegundos.

Tabela Forçar Saídas:



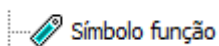
As saídas podem ser forçadas para 1, o estado forçado se mantém até mesmo quando o Nexo está em modo STOP.

6.4.4. Bloco de dados



O bloco de programa possui três partes, nomeadas de, MAIN (Programa principal), INT-1 (rotina de interrupção) e SBR-0 (sub-rotina). Selecione qualquer um deles e click com o direito para remover ou adicionar um novo INT ou SBR. O MAIN não pode ser removido.

6.4.5. Bloco de dados



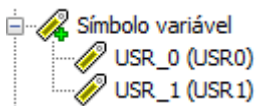
De um duplo clique em "Símbolo função" para abrir a interface:

		símbolo	endereço	comentário
	✓	MAIN	INT0	
	✓	INT_1	INT1	
	✓	SBR_0	SBR0	

Você pode mudar os símbolos, endereços, e comentários.

		símbolo	endereço	comentário
	✓	Mk7	INT0	Variável de contagem
	✓	MTX	INT1	Principal
	✓	ALN	SBR0	Teste

6.4.6. Símbolo de variável



De um duplo clique para abrir:

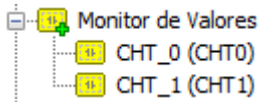
		símbolo	endereço	Tipo de da...	comentário
				BOOL	
				BOOL	
				BOOL	
				BOOL	

É possível alterar os símbolos, endereços tipos e comentarios:

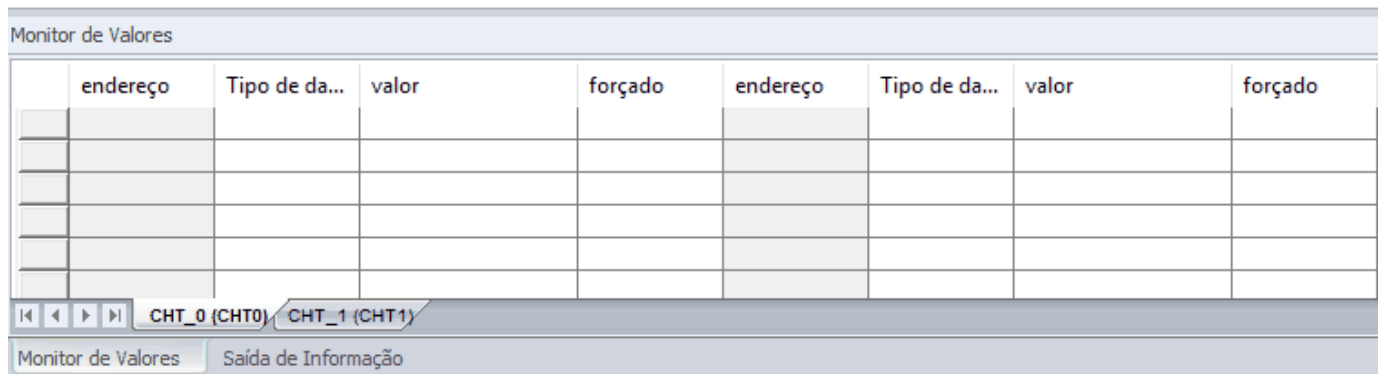
		símbolo	endereço	Tipo de da...	comentário
	✓	Iniciar	I0.0	BOOL	Início do processo
	✓	Parar	I0.1	BOOL	Parar processo
		Alarme	Q0.0	BYTE	Alarme
				BOOL	

Quando o endereço e o tipo não coincidem o endereço fica vermelho.

6.4.7. Monitor de valores

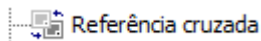


Duplo clique para abrir:

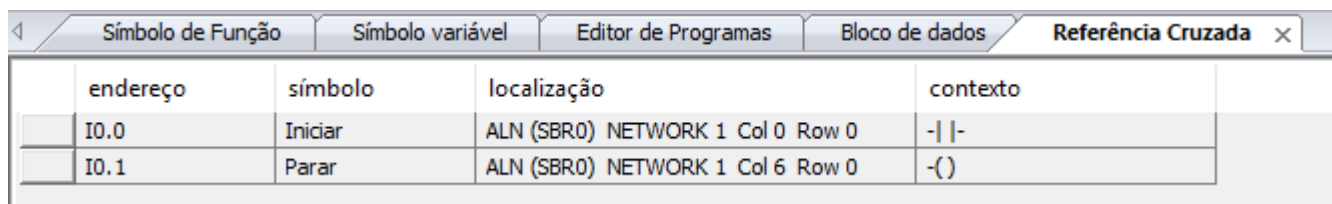


No Monitor de valores você pode monitorar e forçar valores.

6.4.8. Referência cruzada

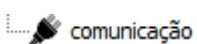


A referência cruzada mostra o endereço, símbolo, localização, e instrução como na imagem abaixo:



Não é preciso fazer uma compilação para usar a referência cruzada.

6.4.9. Referência cruzada



Configuração de interface de comunicação com o Nexó:

comunicação

Porta serial Modbus TCP/IP CAN Kvaser CANpro CANalyst-II

Padrão

Estação: 0

Porta:

Parâmetros do barramento

Taxa de: 9600 bps

Paridade: NONE

Bit de: 1 Bit

OK Cancelar

Porta Serial: Configuração serial padrão.

MODBUS TCP/IP: Ainda não é suportado.

CAN Kvaser, CANpro, CANalyst-II são modos de comunicação CAN.

MODBUS TCP/IP: Ainda não suportado.

comunicação

Porta serial Modbus TCP/IP CAN Kvaser CANpro CANalyst-II

Padrão

Estação: 0

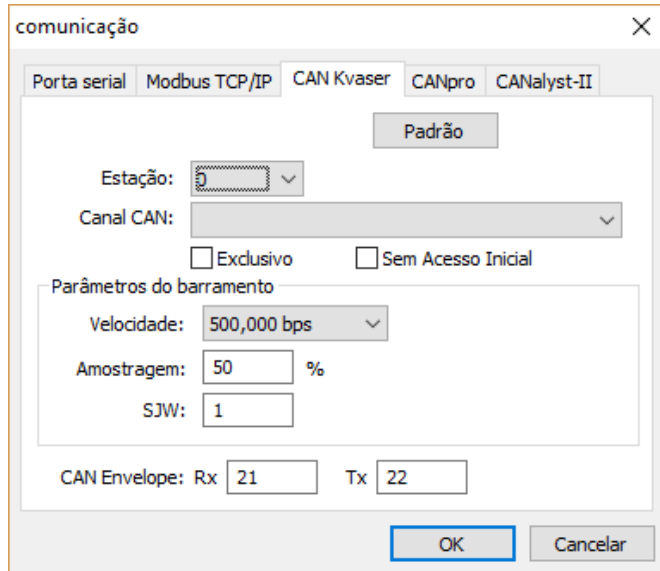
Adaptador: 0.0.0.0 (Qualcomm Atheros AR8172/8176/8178 PC)

Endereço IP: 192.168.0.191

Porta: 8888

OK Cancelar

CAN Kvaser, CANpro, CANalyst-II são modos de comunicação CAN.



6.4.10. Editor de Programas

Símbolo de Função		Símbolo variável		Editor de Programas	
	símbolo	Tipo de var...	Tipo de da...	comentário	
		TEMP	BOOL		
		TEMP	BOOL		
		TEMP	BOOL		
		TEMP	BOOL		

LINHA 0
 Início do Programa

```

    M0.0 ---| |--- Q0.0
                |
                |--- Q0.1
    
```

LINHA 1

```

    M0.1 ---| |--- V0.0
                |
                |--- S
                |
                |--- 10
    
```

LINHA 2

◀ ▶ ⏪ ⏩
 Mk7 (INT0) | MTX (INT1) | ALN (SBR0)

6.4.11. Saída de Informação

```

Saída de Informação
Compilando bloco ...
Mk7 (INT0)
MTX (INT1)
MTX (INT1), NETWORK 0 Erro : Erro de análise Bloco(instrução)
ALN (SBR0)
total 1 Erro(s)
Compilando blocos de dados ...
DAT_0 (DAT0)
DAT_1 (DAT1)
Tamanho do bloco de dados : 0 Byte(s) (0.0%)
Compilando o bloco de sistema ...
SYS_0 (SYS0)
SYS_1 (SYS1)
SYS_2 (SYS2)
SYS_3 (SYS3)
SYS_4 (SYS4)
SYS_5 (SYS5)
SYS_6 (SYS6)
SYS_7 (SYS7)
SYS_8 (SYS8)
SYS_9 (SYS9)
Tamanho do bloco de sistema : 150 Byte(s) (58.6%)
Compilando bloco ...
Mk7 (INT0)
MTX (INT1)
ALN (SBR0)
Tamanho do bloco : 4 + 184 Byte(s) (0.8% + 0.3%)
total 0 Erro(s)
Monitor de Valores Saída de Informação
    
```

Status: Capítulo 5.3.7.

Informações: A janela de informações mostra erros de compilação, informações sobre memória, bloco de sistema, etc.

6.5. Conceitos de programação

6.5.1. Como o programa funciona

O programa roda em um ciclo, o Nexo lê e escreve os dados continuamente. Quando é feito o download do programa para o Nexo e colocado em modo RUN, a unidade de processamento central (CPU) executa o programa na seguinte ordem:

- A: Leitura das entradas.
- B: Uso das entradas para o controle lógico.
- C: Armazena a imagem do resultado lógico em um registrador.
- D: No fim do programa, envia a imagem do registrador para as saídas.
- E: Repete todos os passos.

O NeXo executa uma serie de tarefas repetidamente. O ciclo de execução dessas tarefas e chamado de scan cycle. As tarefas são:

A: Leitura do status de entrada.

B: Execução das instruções do programa, armazenamento de dados.

C: Execução das comunicações.

D: Execução de auto teste e diagnostico de CPU. O Nexo confirma se o hardware, a memória de programa e os módulos de expansão estão em operação normal.

E: Os valores armazenados na imagem são escritos na saída.

Atenção: A execução do scan cycle só ocorre quando o Nexo está em modo RUN.

6.5.2. Visão geral endereçamento

Identificando endereços absolutos e simbólicos

Você pode usar endereços absolutos ou símbolos para identificar instruções no programa. A referência absoluta usa a área de memória (bit/byte) para identificar os endereços. A referência simbólica usa letras, números e alguns caracteres para identificar endereços ou valores.

Como mostrar o endereço no editor does programa:

I0.0 – O endereço absoluto é composto da area de momoria mais o bit endereço.

#INPUT1 – O simbolo ‘#’ indica uma variavel local.

INPUT1 – Nomenclatura global

??.? ou ???? – Indica um endereço indefinido (deve ser definido antes da compilação).

Símbolo Global

Os símbolos do escopo Global podem ser usados em todo o programa.

Pela tabela global de variáveis, você pode assimilar um símbolo a uma variável.

Variável Local

As variáveis locais são declaradas dentro de uma POU, e só podem ser usadas na POU onde foi declarada

Atenção: Se você usar um mesmo símbolo no escopo local e no global, as variáveis locais têm prioridade.

As variáveis locais usam a memória temporária L, não utilizando assim memória de programa no CLP.

As sub-rotinas que usam apenas variáveis locais, são rotinas moveis, que podem ser usadas em mais de um programa. Como a variável local é temporária, lembre-se sempre de inicializar os valores na POU. Somente a tabela de variáveis global suporta variáveis constantes.

6.5.3. Como organizar o programa

Elementos básicos de controle do programa

O programa NeXo consiste nos seguintes tipos de programa:

Programa Principal: Corpo principal do programa, é onde colocamos as instruções de controle. As instruções são executadas em sequência uma vez a cada ciclo (scan).

sub-rotina: a sub-rotina é executada quando é chamada pelo main, interrupt, ou até mesmo outra sub-rotina.

interrupt routine Sub-rotina de interrupção, é executada somente quando o evento de interrupção ocorre.

Como finalizar uma POU

O compilador usa funções incondicionais para terminar cada POU: END, MEND, RET ou RETI. Se alguma dessas funções incondicionais forem usadas, o compilador retornará uma mensagem de erro.

6.6. Como usar o Nexo Ladder

6.6.1. Como contruir um novo projeto


Clique , Criar novo projeto.

Abra um projeto existente


Clique em no ícone "Arquivo", selecione "Abrir" ou "Abrir arquivo PWM".

6.6.2. Principio de trabalho em Ladder

A lógica Ladder (LAD) é uma linguagem gráfica similar a um diagrama elétrico onde vários componentes gráficos compõem uma rede lógica em forma de escada. Esses componentes são:

Contato : Quando o valor logico do contato NA é igual a 1 o do contato NF é igual a 0, a 'energia' pode passar por esses contatos.

Bobina : A bobina representa uma saída.

Bloco  : Cada bloco representa uma função.

A "energia" flui do lado esquerdo da rede até os blocos ou bobinas no lado direito.

6.6.3. Regras de paralelos e series

Uso dos contatos

Cada Linha deve começar com um contato. E não pode terminar com um contato.

Uso das bobinas

A linha não pode começar com uma bobina.

A bobina é usada para terminar uma linha lógica. Uma linha lógica pode ter mais de uma bobina, que pode ser inserida em uma ramificação paralela a linha lógica.

As bobinas não podem ser inseridas em série.

Uso dos Blocos de Função

Se o bloco possuir ENO, o bit habilitado, será estendido até a saída do bloco; Significa que você poderá inserir vários blocos em série desse. Se o bloco não possuir ENO não poderá ser utilizado mais blocos na mesma linha lógica.

Tamanho limite da Linha logica

A área onde é colocada o elemento lógico é chamada Célula. Uma linha lógica pode possuir até 32 células verticais e 32 horizontais.

6.6.4. Entrada de comandos

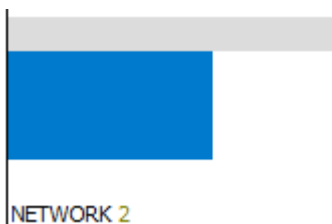
Linha

Você pode usar linhas verticais e horizontais para ligar os componentes em uma Linha Lógica.

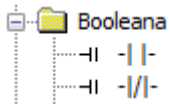


Duplo clique na Raiz do projeto

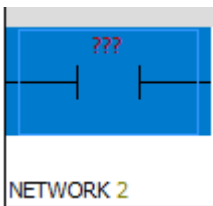
1. Coloque o cursor na célula onde deseja inserir o componente lógico.



2. Selecione a instrução desejada com 2 cliques.

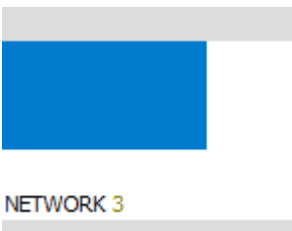


A instrução irá aparecer na célula selecionada:



Pela barra de ferramentas ou atalhos

1. Selecione a célula.

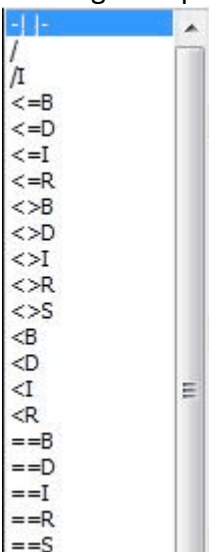


2. Clique no botão desejado na barra de ferramentas



Ou use os atalhos do teclado (F4= contato, F6= bobina, F9= bloco).

3. O segundo passo é selecionar a instrução na lista que irá aparecer.



6.6.5. Como endereçar os elementos

Quando um elemento lógico é inserido, a instrução fica com sinais de interrogação. Isso indica que a instrução não foi vinculada a nenhum endereço de memória e o compilador retornará um erro. Você poderá inserir um endereço de memória, ou um símbolo já declarado na tabela global ou local.

Atenção: Quando um símbolo é declarado na tabela local, automaticamente é vinculado a uma memória L.

6.6.6. Como editar os elementos

Cortar, copiar, colar, ou apagar múltiplas linhas

Clicando e arrastando, ou clicando com a tecla Shift pressionada é possível selecionar múltiplas linhas.

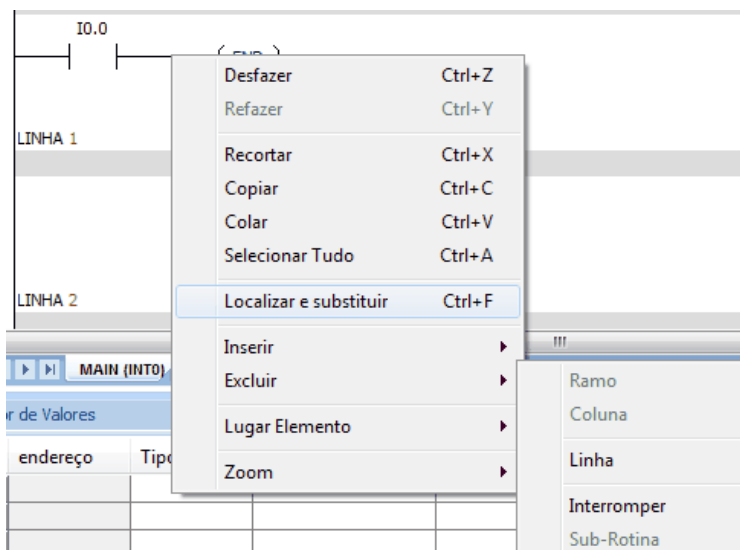
É possível cortar e copiar:

1. Textos do programa.
2. Instruções em LAD e STL.
3. Uma linha logica inteira.
4. Múltiplas linhas adjacentes
5. Todas as linhas
6. Tabela de símbolos, Linha e coluna da tabela
7. Tabela de monitoramento, linha e coluna da tabela.

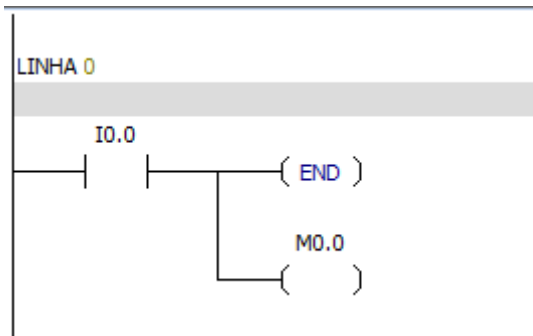
Editar células, instruções, endereços e ramificações

Selecione a célula, e clique com o botão direito do mouse:

É possível apagar, cortar, copiar e colar elementos e ramificações:

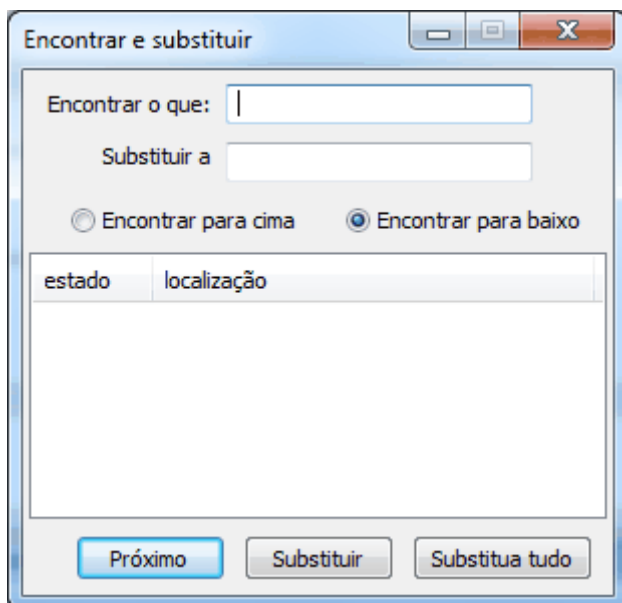


Use também as teclas DELETE ou BACKSPACE para apagar células.
Use o mouse para selecionar as linhas verticais.



6.6.7. Como usar o Encontrar / Substituir

1. Selecione Editar > Encontrar e Substituir, ou o atalho Ctrl+F:



Função Encontrar

1. Entre com a string para procurar, selecione entre Encontrar para cima ou Encontrar para baixo.

Função Substituir

1. Entre com a string para procurar, selecione entre Encontrar para cima ou Encontrar para baixo.
2. Entre com a string para substituir no campo "Encontrar o que".
3. Clique em "Próximo" para encontrar a próxima string.
4. Para substituir, clique "Substituir". Se quiser substituir todos, clique "Substituir todos".

Onde usar

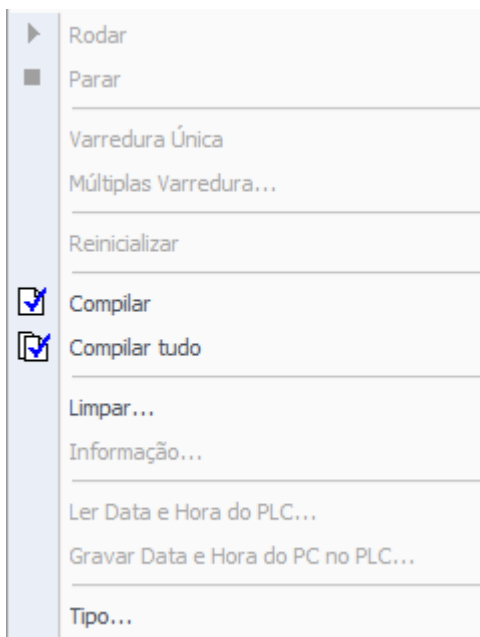
Você pode usar o "Encontrar" e "Substituir" na janela de edição do programa para localizar strings como instruções, número de linha, títulos de linhas. O "Encontrar" não localiza strings em comentários.


6.6.8. Como identificar erros no programa


Os erros aparecem em vermelho. Quando um valor ou símbolo com erro é substituído a fonte volta para a cor normal automaticamente.

6.6.9. Como compilar

Você pode usar o botão na Barra de Ferramentas "CLP" ou menu "Compilar".



"Compilar"  Permite compilar um único elemento do projeto. Quando você seleciona "compilar", a janela atual é compilada e as outras janelas não são compiladas.

"Compilar tudo"  Compila o editor do programa, o bloco do sistema e o bloco de dados. Quando você usa o comando "Compilar tudo", todas as janelas são compiladas.

Use a Saída de Informação para resolver problemas

Quando um programa é compilado, a janela de saída lista todos os erros do programa. Os erros incluem localização (Network, linha e coluna) e tipo de erro.

6.6.10. Como salvar o projeto

Você pode usar o botão "Salvar" na barra de ferramentas para salvar o seu projeto ou usar o atalho CTRL + S.



"**Salvar**" permite que você salve todas as mudanças rapidamente no seu projeto.

"**Salvar como**" permite que você altere o nome do projeto atual e o local do diretório.

6.7. Comunicação e transferência do programa

6.7.1. Configuração de comunicação

Para estabelecer uma conexão entre o computador e o Nexo é preciso selecionar o tipo de conexão de acordo com o cabo utilizado.

Você pode mudar os tipos de conexão em qualquer momento.

Passos para estabelecer a comunicação:

1. Use o cabo de comunicação para conectar o CLP e o computador.

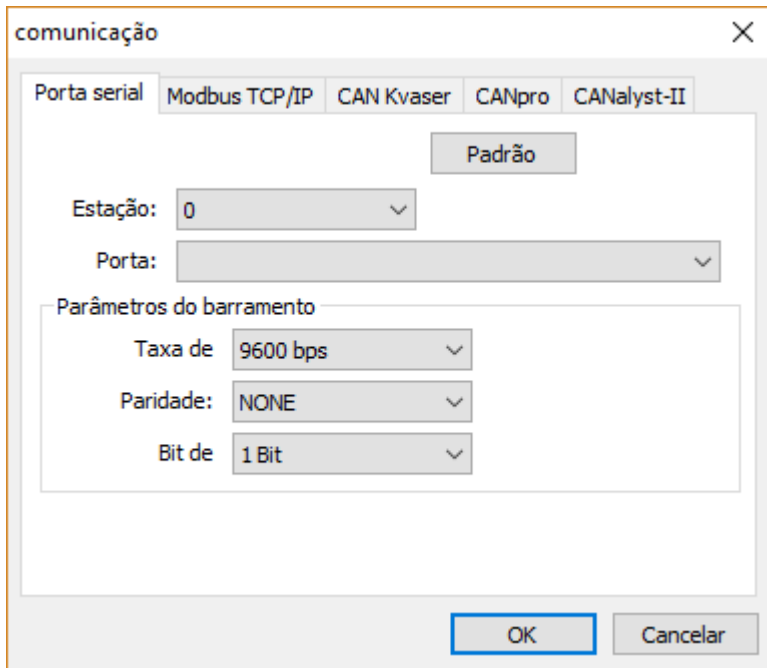
Parâmetros Padrão:

Estação: 0

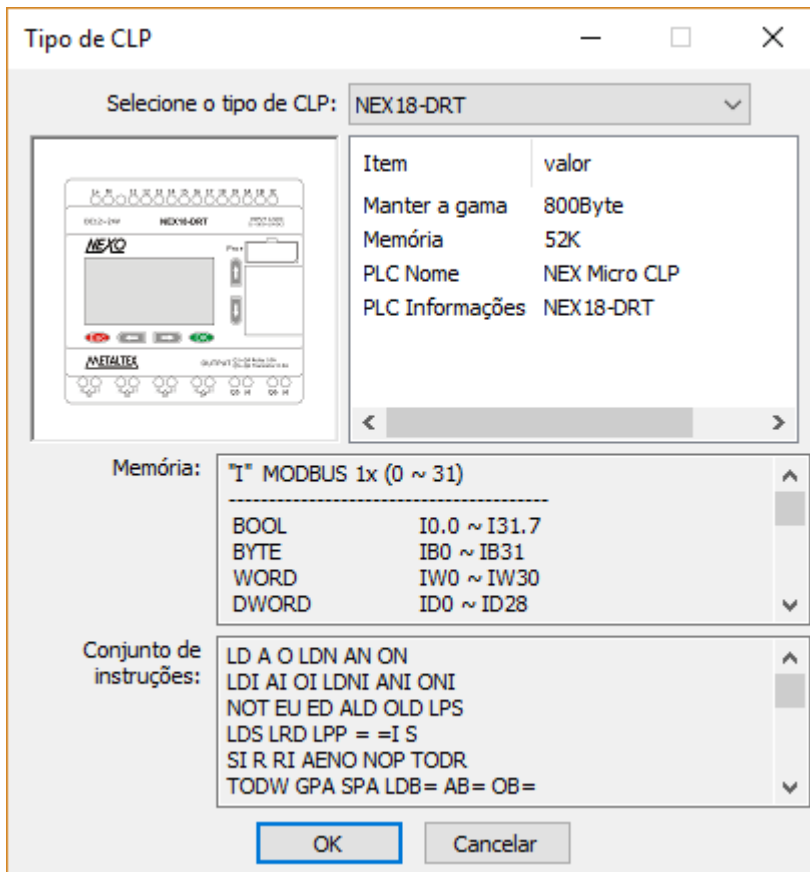
Porta: Selecione a porta correta

Taxa de: 9600 bps

Stop bit: 1 bit





2. Selecione o modelo do CLP: Certifique-se de que o modelo PLC no software seja compatível com o modelo PLC atual.




6.7.2. Download do programa

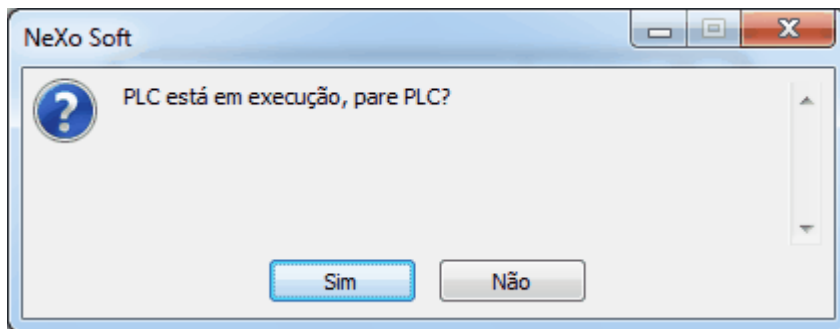
Com a comunicação entre o NexoLadder e o CLP Nexo estabelecida, você pode descarregar o programa, com os seguintes passos:

Atenção: o novo programa subscreve o programa antigo.

1. Antes de descarregar o programa no CLP, o programa deve ser compilado  .

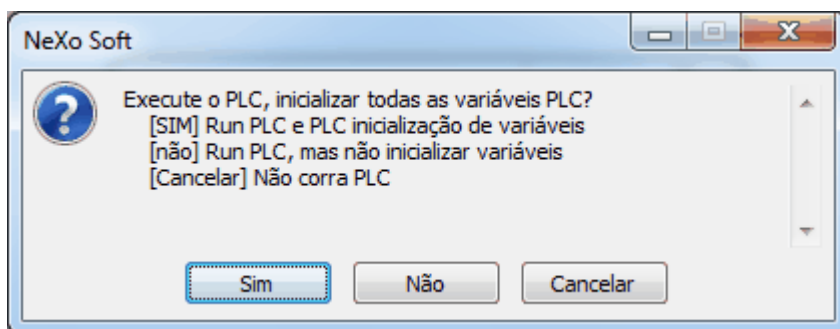
2. Após compilado corretamente, clique no botão "Download"  na barra de ferramentas ou selecione Arquivo > Download.

A interface é a seguinte:






Clique em Sim, o software irá descarregar automaticamente o programa, o bloco de dados e as configurações da CPU para o PLC.

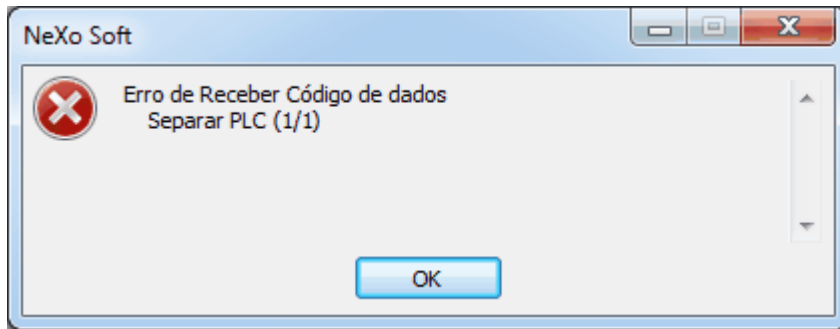
3. Quando o programa é descarregado corretamente, a interface é a seguinte:



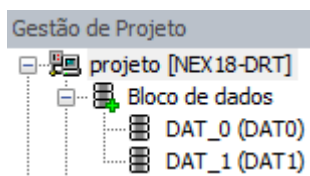
Há três opções para você escolher.

 para monitorar o programa. Quando você escolhe "cancelar", o PLC para. Você pode clicar no botão "executar"  e em seguida, clicar em "conexão"  para monitorar o programa.

5. Se o CLP definido no software for diferente do PLC da conexão real, o software exibirá uma mensagem de aviso.



6. Você pode clicar duas vezes em "Gestão de Projeto" para modificar o modelo do CLP.



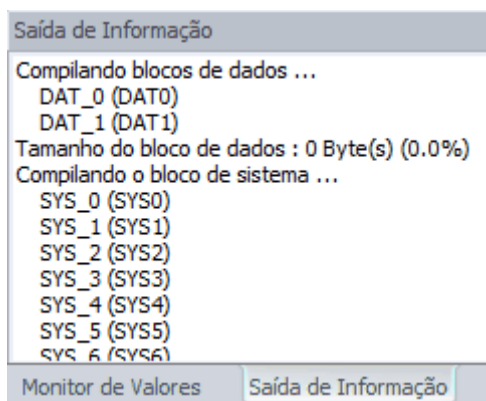
7. Clique no botão "download" para descarregar o programa novamente.

8. Se o programa for transferido corretamente, você pode alterar o CLP do modo "Parado" para o modo "Execução".

6.7.3. Como corrigir erros de compilação e erros de download

A janela de saída exibe automaticamente informações do programa e mensagens de erro quando você compila um programa ou descarrega um programa.

A informação normalmente inclui o erro de Network, a posição da coluna, linha, código de erro e as instruções.



Se você fechou a janela de saída, selecione Vista > Quadro > saída de informação na barra de menu, para exibir novamente a janela de saída.

6.7.4. Como monitorar e depurar o programa

Após o download do programa, você pode usar a função de diagnóstico da barra de ferramentas "debug".

Barra de ferramentas Depurar : 

O que é "estado de monitoramento"?

O estado de monitoramento mostra o valor atual dos dados do CLP e as informações do estado atual. Você pode monitorar, ler, escrever e impor os valores dos dados do CLP usando a tabela de status. Quando o programa é executado, existem duas maneiras de visualizar o CLP Dinâmica de dados.

Monitoramento de tabela Exibe o status dos dados na tabela: você pode especificar o endereço, tipo de dados, valor e forçado.

Monitoramento do status do programa Exibe o status dos dados na janela do editor do programa: o valor atual do dado do PLC é exibido na instrução STL ou no gráfico LAD.


As condições do status de exibição de dados

1. Comunicação entre o Nexoladder e CLP.
2. Descarregue o programa para o CLP.
3. Para visualizar as mudanças contínuas do estado de dados do CLP, o mesmo deve estar em modo RUN.
4. Se o programa que você monitorar não for implementado, não haverá um estado exibição.

Atenção:

Ao transferir o programa, você deve selecionar o modo RUN no CLP para executar o programa. Por exemplo, no modo STOP, você não poderá ver os resultados esperados da operação lógica do programa.

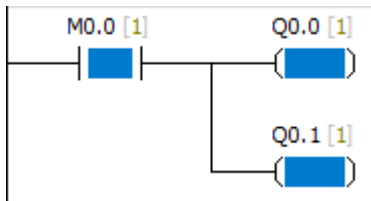
Como visualizar o status dos dados

No modo RUN, clique na conexão  para monitorar o programa. insira o endereço dos dados que deseja visualizar na tabela de estados, a tabela de status mostrará seu valor atual.

A cor do status de execução:

Contato: quando o contato é ligado, as instruções mudarão a cor.

Bobina: quando a saída é ligada, as instruções mudarão a cor.



Os valores do estado são coletados de maneira contínua ou instantânea

Continuidade

1. Abra a janela do editor do programa e inicie o "monitoramento do status do programa". Quando o CLP estiver no modo RUN, você pode visualizar o estado contínuo dos dados do programa.
2. Abra a janela da tabela de status e inicie o "monitoramento da tabela de status". Quando o CLP estiver no modo RUN, você pode visualizar o estado contínuo dos dados do programa.

Instantâneo

Com o CLP no modo STOP, você pode coletar uma única atualização de status, você também pode usar as funções de "varredura múltipla" e "digitalização única".

CLP modo RUN / STOP

Segue abaixo os seguintes métodos para alterar o modo de operação do CLP:

1. Clique no botão "RUN" para executar o modo RUN. Ou clique no botão "STOP" para executar o modo STOP.
2. Selecione o comando CLP > executar, para executar o modo RUN, ou selecione o comando PLC > stop, para executar o modo STOP.
3. Insira uma instrução no programa (modo STOP).

Atenção:

Quando o CLP estiver no modo STOP, você poderá executar as seguintes operações:

1. Use a tabela de status ou a janela de monitoramento do status do programa para ver o valor atual dos dados.
2. Fornecer um número limitado de varreduras.

Quando o CLP estiver no modo RUN, você não poderá usar a função "primeira verificação" ou "varredura múltipla". Quando o CLP estiver no modo RUN, você poderá escrever e forçar dados na tabela de status. Você também pode executar as seguintes operações:

1. Utilize a tabela de status para visualizar o estado contínuo dos dados do programa.
2. Use o monitoramento do status do programa para visualizar o estado contínuo dos dados do programa.

Prioridade e cancelar prioridade

Forçado Digite o endereço e o valor que deseja forçar na tabela de estados. Em seguida, selecione a função "aplicação". Antes de cancelar o obrigatório, a função aplicação foi efetiva.

Monitor de Valores

	endereço	Tipo de da...	valor	forçado	endereço	Tipo de da...	valor	forçado
	M0.0	BOOL	1		I0.0	BOOL	0	não é o...
	VB0	BYTE	16#20					aplicação
	VB1	BYTE	16#20					não é obrigatória
	VB2	BYTE	16#20					desconhecido
	VB3	BYTE	16#20					
	VB4	BYTE	16#2F					

Monitor de Valores Saída de Informação

Função "Aplicação" abrange as funções "ler imediatamente" e "escrever imediatamente". Os pontos de E / S podem ser forçados e outras áreas de armazenamento não podem ser forçadas.

Cancelar o obrigatório Selecione "não forçado" na tabela de status para cancelar obrigatório

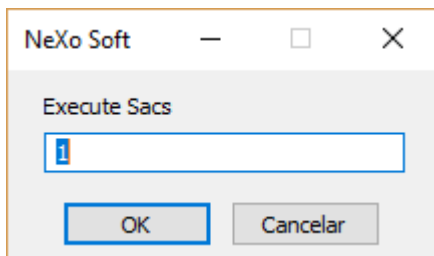
Como executar um número limitado de varreduras

Única varredura:

1. PLC deve ser selecionado para o modo PARAR.
2. Selecione PLC > única varredura na barra de menu..

Múltiplas varreduras :

A caixa de diálogo aparece da seguinte maneira:



3. Entre com o número de varreduras em seguida clique em "OK".

6.7.5. Operação e opções do CLP

Elementos do programa de controle

Programa Ladder

No LAD, os elementos básicos da lógica são representados por contatos, bobinas e caixas.

A entrada é representada por um símbolo chamado contato. O contato é dividido em contato normalmente aberto e contato normalmente fechado.

Contato normalmente aberto: é um contato aberto.

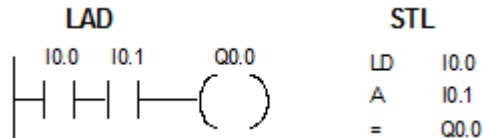
Contato normalmente fechado: é um contato fechado.

A saída é representada por um símbolo chamado bobina.

Os blocos são blocos de função com várias funções. Os blocos podem facilitar a programação.

Programa STL

Os elementos do programa STL são representados por instruções. O diagrama de linha e as instruções são seguintes:



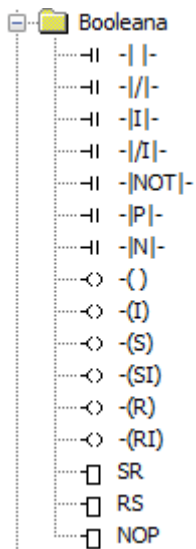
Configuração dos blocos do sistema

Instrução: a anotação detalhada da interface esta em operação ---> blocos do sistema

Instructions : The detailed annotation of operation interface--->System blocks

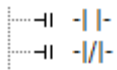
7. Descrições de instruções Nexo Ladder

7.1. Boleana



[Como chamar o arquivo de ajuda](#)

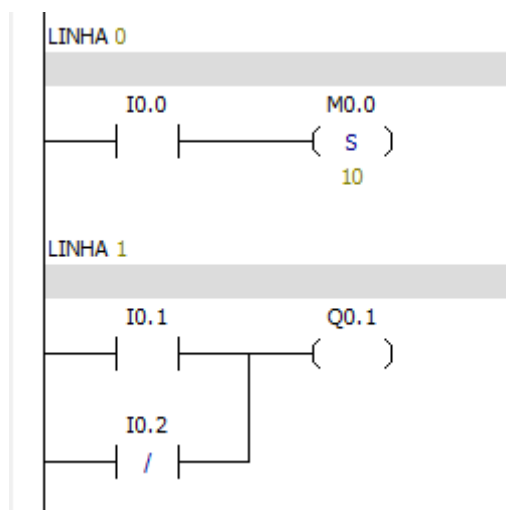
7.1.1. Normalmente aberto e Normalmete fechado



Entrada / Saída	Operando
Bit (LAD, STL)	I, Q, M, SM, T, C, V, S, L
Entrada (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit

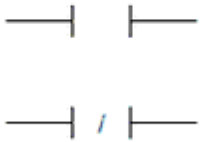
Normalmente aberto e Normalmente fechado	
	Quando o bit for igual a 1, o contato normalmente aberto (NA) é fechado e o normalmente fechado (NF) é aberto.
	Quando o bit for igual a 0, o contato NA permanece aberto e o contato NF permanece fechado.
	Em STL, o contato NA é representado pelas instruções "LD", "E" e "OR".
	Em STL, o contato NF é representado pelas instruções "NOT", "NOT AND" e "NOT OR".

Exemplo:

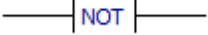


[Como chamar o arquivo de ajuda](#)

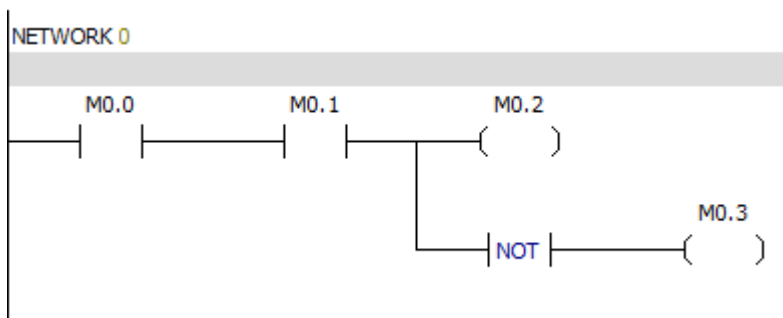
7.1.2. Imediatamente o contato NA e imediatamente o contato NF

<p>Imediatamente o contato NA imediatamente o contato NF</p> 	<p>Quando o CLP executa as instruções, a instrução imediata obtém o valor de entrada real, mas o CLP não atualiza o registro da imagem do processo. A atualização imediata do contato não depende do ciclo de verificação do CLP; Será atualizado imediatamente.</p> <p>Quando o ponto de entrada real é 1, imediatamente o contato NA é fechado.</p> <p>Quando o ponto de entrada real é 0, imediatamente o contato NF é fechado.</p> <p>Em LAD, as instruções imediatas de contato NA e contato NF são representadas pelos contatos.</p>
--	--

7.1.3. NOT inverte instrução

<p>Instrução NOT</p> 	<p>As funções da "instrução NOT" são as seguintes:</p> <p>Quando a entrada é 0, a saída é 1.</p> <p>Quando a entrada é 1, a saída é 0.</p> <p>Em LAD, a instrução NOT é representada por um contato.</p>
---	--

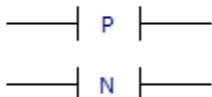
Exemplo:



7.1.4. Borda de subida e borda de descida

Entrada / Saída	Operando	Tipo de dado
Entrada (FBD)		Booleana
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana

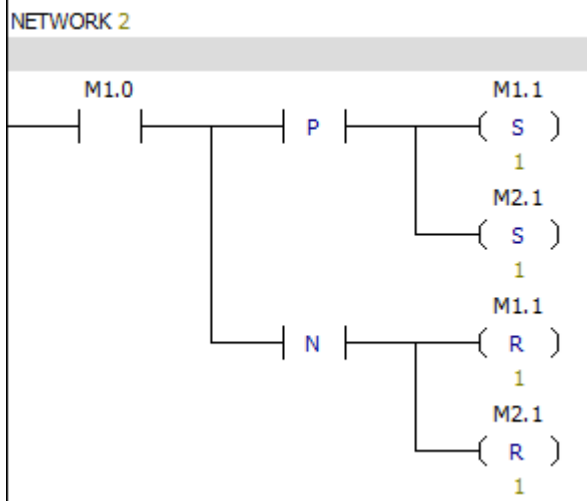
Borda de subida e Borda de descida



Quando a lógica é convertida de 0 a 1, o tempo de condução dos contatos da borda de subida é um ciclo de varredura.

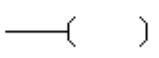
Em LAD, a borda de subida e a borda de descida são representadas pelos contatos.

Exemplo:

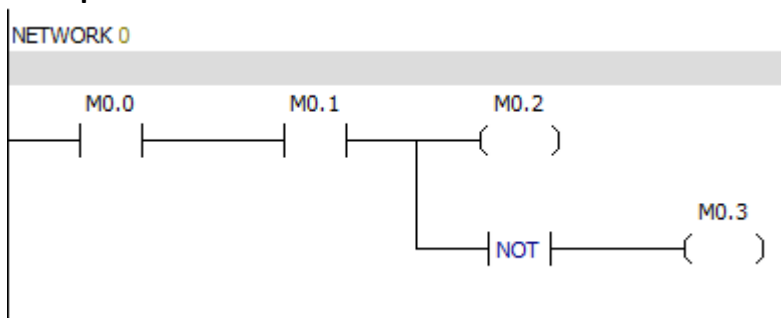


7.1.5. Saída

Entrada / Saída	Operando	Tipo de dados
Bit	I, Q, M, SM, T, C, V, S, L	Booleana
Entrada (LAD)	Enable bit	Booleana
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana

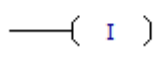
<h2>Saída</h2> 	<p>A instrução de saída grava o novo valor do bit de saída para processar o registro de image.</p> <p>Em LAD e FBD, quando a instrução de saída é executada, o CLP irá abrir ou fechar o bit de saída no registro da imagem do processo.</p>
--	--

Exemplo:



7.1.6. Saída imediata

Entrada / Saída		Tipo de dados
Bit	Q	Booleana
Entrada (LAD)	Enable bit	Booleana
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana

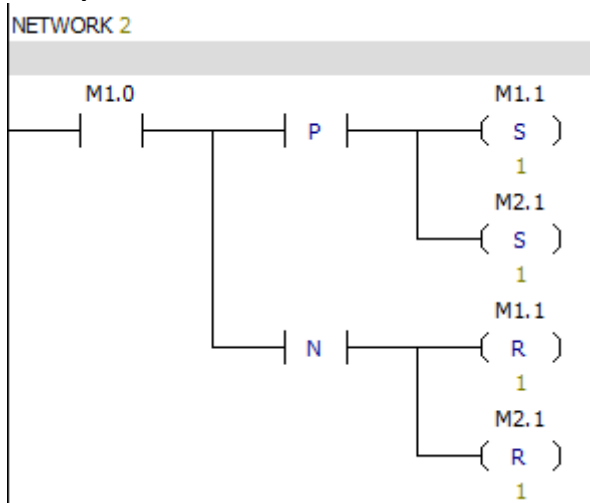
<h2>Saída imediata</h2> 	<p>O novo valor gerado pela execução da instrução de saída imediata é gravado na saída real e no registro de imagem do processo correspondente.</p>
---	---

7.1.7. Set e Reset

Entrada / Saída	Operando	Tipo de dados
Bit	I, Q, M, SM, T, C, V, S, L	Booleana
N	VB, IB, QB, MB, SMB, SB, LB, AC, constant, *VD, *AC, *LD	Byte

Set e Reset	
—(S)	A instrução "Set" pode mudar um ou mais bits para 1.
—(R)	A instrução "Reset" pode mudar um ou mais bits para 0.
	O valor de N está entre 1 e 255.

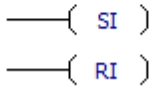
Exemplo:



7.1.8. Set e Reset imediato

Entrada / Saída	Operando	Tipo de dado
Bit	Q	Booleana
N	VB, IB, QB, MB, SMB, SB, LB, AC, constant, *VD, *AC	Byte

Set e Reset imediato



"Set imediato" pode definir muitos pontos imediatamente.

"Reset imediato" pode redefinir muitos dos pontos imediatamente.

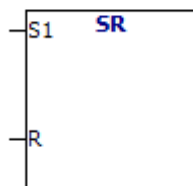
O valor de N é entre 1 e 128.

"I" significa "referência imediatamente"; o novo valor gerado pela execução da instrução é escrito para a saída real e o registro de imagem de processo correspondente.

7.1.9. Instrução SR

Entrada / Saída	Operando	Tipo de dado
S1, R (LAD)	Enable bit	Booleana
S1, R (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
Saída (LAD)	Enable bit	Booleana
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
xxx	I, Q, M, V, S	Booleana

Instrução SR

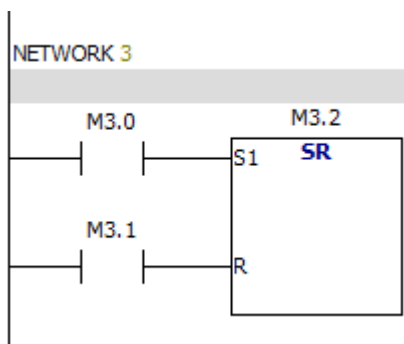


Quando R e S1 são iguais a 1, a saída é 1.

Segue a tabela verdade da instrução "SR":

Instrução	S1	R	Saída
SR	0	0	Estado Anterior
	0		0
	1	0	1
	1	1	1

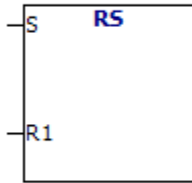
Exemplo:



7.1.10. Instrução RS

Entrada / Saída	Operando	Tipo de dado
S, R1 (LAD)	Enable bit	Booleana
S, R1 (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
Saída (LAD)	Enable bit	Booleana
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
xxx	I, Q, M, V, S	Boolean

Instrução RS

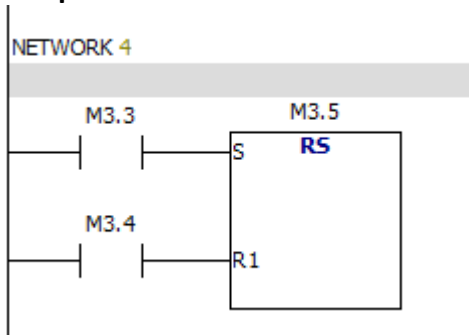


Quando R1 e S são iguais a 1, a saída é 1.

Segue a tabela verdade da instrução "RS":

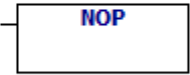
Instrução	S	R1	Saída
RS	0	0	Estado anterior
	0	1	0
	1	0	1
	1	1	0

Exmeplo:



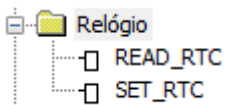
7.1.11. Instrução NOP

Instrução NOP



A instrução NOP é inválida na execução do programa do usuário. A instrução NOP não pode ser usada no modo FBD. O valor de N está entre 0 e 255

7.2. Instrução Relógio

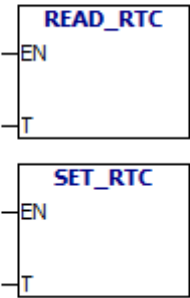


[Como chamar o arquivo de ajuda](#)

7.2.1. Lê e grava a hora em tempo real

Entrada / Saída	Operando	Tipo de dado
T	VB, IB, QB, MB, SMB, SB, LB, *VD, *AC, *LD	Byte

Lê e grava a hora em tempo real



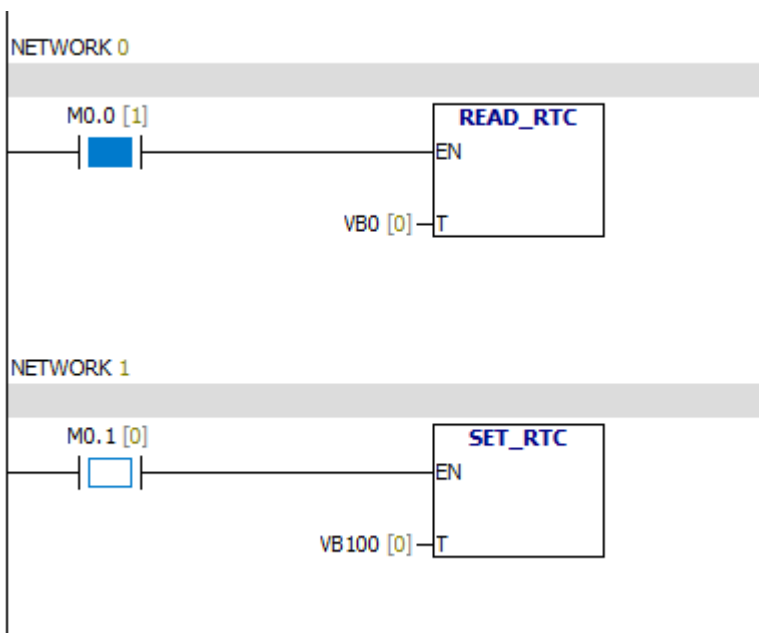
A instrução READ-RTC lê a hora e data atuais do relógio e carrega no buffer de tempo de 7 bytes começando no endereço T.

A instrução SET-RTC grava a hora e a data atuais para o relógio especificado pelo buffer de tempo de 7 bytes começando no endereço T.

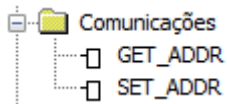
Todos os valores de data e hora devem ser codificados no formato USINT. Consulte a tabela a seguir:

T Byte	Direção	Tipo de Byte
0	Segundo	USINT
1	Minuto	USINT
2	Hora	USINT
3	Data	USINT
4	Semana	USINT
5	Mês	USINT
6	Ano	USINT

Exemplo:



7.3. Comunicação



[Como chamar o arquivo de ajuda](#)

7.3.1. Ler endereço da porta

Entrada / Saída	Operando	Tipo de dado
ADDR	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	Byte
PORT	Constante (0 ou 1)	Byte

Endereço de Porta

GET_ADDR

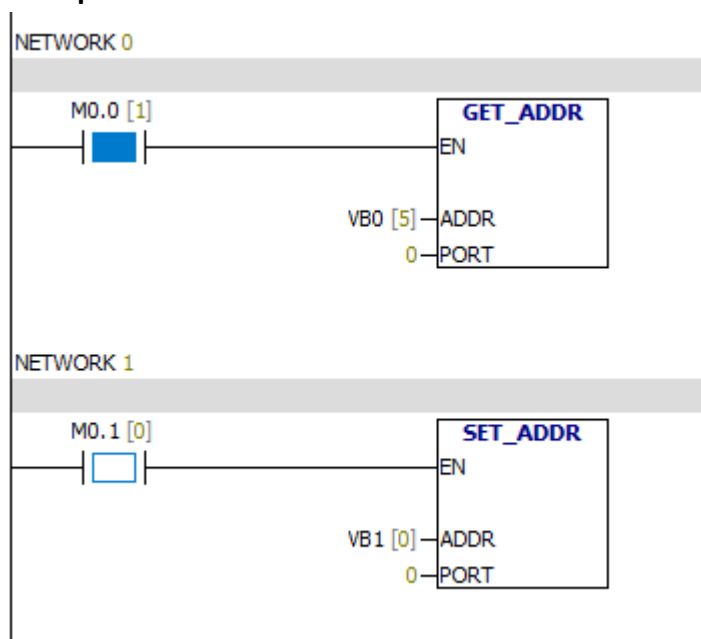
—EN

—ADDR

—PORT

A instrução GET -ADDR lê a porta do CLP e coloca o valor no endereço especificado no ADDR.

Exemplo



7.3.2. Definir endereço da porta

Entrada / Saída	Operando	Tipo de dado
ADDR	VB, IB, QB, MB, SB, SMB, LB, AC, Constant, *VD, *LD, *AC	Byte
PORT	Constant (0 or 1)	Byte

Set port address

SET_ADDR

—EN

—ADDR

—PORT

A instrução SET- ADDR define a porta para o valor especificado no ADDR.

O novo endereço não é salvo permanentemente.

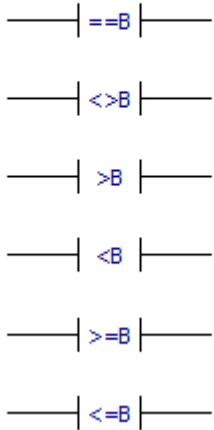
7.4. Comparação

- Comparação
- I -|=B|-
- I -|<>B|-
- I -|>B|-
- I -|<B|-
- I -|>=B|-
- I -|<=B|-
- I -|=I|-
- I -|<>I|-
- I -|>=I|-
- I -|<=I|-
- I -|>I|-
- I -|<I|-
- I -|=D|-
- I -|<>D|-
- I -|>=D|-
- I -|<=D|-
- I -|>D|-
- I -|<D|-
- I -|=R|-
- I -|<>R|-
- I -|>R|-
- I -|<R|-
- I -|>=R|-
- I -|<=R|-
- I -|=S|-
- I -|<>S|-

7.4.1. Comparação de Byte

Entrada / Saída	Operando	Tipo de dado
Entrada	IB, QB, MB, SMB, VB, SB, LB, AC, constant, *VD, *LD, *AC	Byte
Saída (FBD)	I, Q, M, SM, T, C, V, S, L	Boolean

Comparação de Byte



As instruções de comparação de bytes são usadas para comparar dois valores: IN1 e IN2.

A comparação inclui: IN1 >= IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2 ou IN1 <> IN2. Byte comparação sem símbolo.

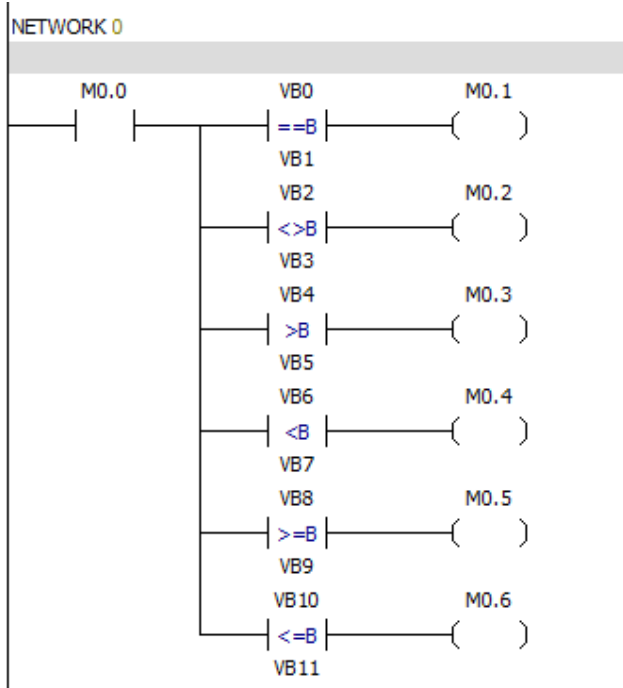
Em LAD, o contato está aberto quando o resultado é 1.

Atenção:

As seguintes condições são erros sérios. Esse erro fará com que o CLP pare imediatamente a execução do programa:

1. Introduzir o endereço indireto.
2. Digitar número real.

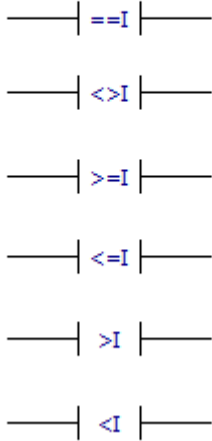
Exemplo:



7.4.2. Comparação de Inteiros

Entrada / Saída	Operando	Tipo de dado
Entrada	IW, QW, MW, SW, SMW, T, C, VW, LW, AIW, AC, Constant, *VD, *LD, *AC	Inteiro
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana

Comparação de Inteiros



As instruções de comparação são usadas para comparar dois valores: IN1 e IN2.

Comparação de números inteiros com símbolo (16 # 7FFF > 16 # 8000).

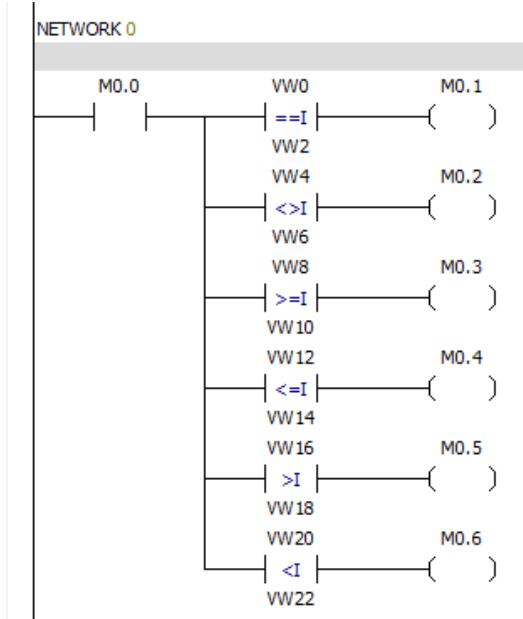
Em LAD, quando o resultado da comparação é verdadeiro, o contato estará aberto.

Na FBD, quando o resultado da comparação for verdadeiro, a saída será aberta.

Atenção: as seguintes condições são erros sérios. Esses erros farão com que o CLP pare imediatamente a execução do programa:

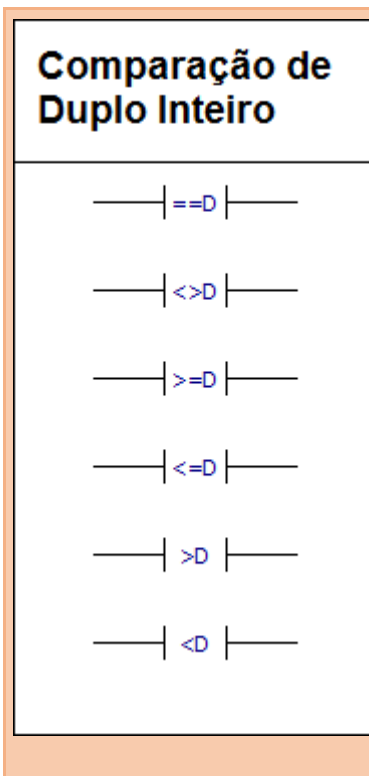
1. Introduzir o endereço indireto.
2. Digitar número inteiro.

Exemplo:



7.4.3. Comparação de duplo inteiro

Entrada / Saída	Operando	Tipo de dado
	ID, QD, MD, SD, SMD, VD, LD, HC, AC, constant, *VD, *LD, *AC	Duplo inteiro
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana



As instruções de duplo inteiro de comparação são usadas para comparar dois valores: IN1 e IN2.

A comparação inclui: IN1 = IN2, IN1 > IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2 ou IN1 <> IN2.

Comparação de número inteiro duplo com símbolo (16 # 7FFFFFFF > 16 # 80000000).

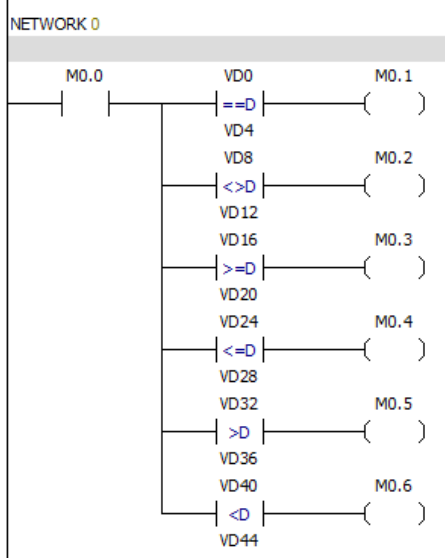
Em LAD, quando o resultado da comparação é verdadeiro, o contato estará aberto.

Na FBD, quando o resultado da comparação for verdadeiro, a saída será aberta.

Atenção: as seguintes condições são erros sérios. Esses erros farão com que o CLP pare imediatamente a execução do programa:

1. Introduzir o endereço indireto.
2. Digitar número inteiro.

Exemplo:



7.4.4. Comparação de número Real

Entrada / Saída	Operando	Tipo de dado
Entrada	ID, QD, MD, SD, SMD, VD, LD, AC, constant, *VD, *LD, *AC	Número Real
Saída (FBD)		Booleana

Comparação de número Real

As instruções de comparação de números reais são usadas para comparar dois valores: IN1 e IN2.

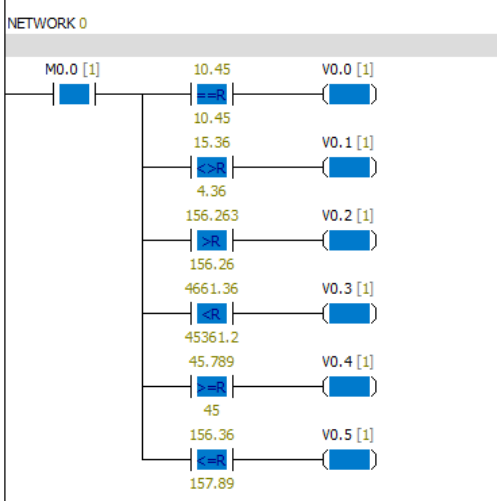
A comparação inclui: IN1 = IN2, IN1 > IN2, IN1 <= IN2, IN1 > IN2, IN1 < IN2 ou IN1 <> IN2. Número Real de comparação com o símbolo.

Em LAD, quando o resultado da comparação é verdadeiro, o contato estará aberto.

Na FBD, quando o resultado da comparação for verdadeiro, a saída será aberta.

2. Digitar número real

Exemplo:



7.4.5. Comparação de String

Entrada / Saída	Operando	Tipo de dado
IN1	VB, Constant string, LB, *VD, *LD, *AC	String
	VB, LB, *VD, *LD, *AC	String
Saída (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana

Comparação de String

As instruções de comparação de String são usadas para comparar duas Strings ASCII: IN1 = IN2, IN1 <> IN2

Em LAD, quando o resultado da comparação é verdadeiro, o contato de comparação será ativado.

O comprimento máximo de uma única String constante é de 126 bytes. O comprimento combinado máximo da String de duas constantes é 242 bytes.

Atenção: As seguintes condições são erros sérios. Esses erros farão com que o CLP pare imediatamente a execução do programa:

1. Introduzir o endereço indireto.
2. Digitar uma string com mais de 254 caracteres de comprimento.
3. O endereço de início e o comprimento da string não podem ser colocados em uma área de memória especificada.

Formato do tipo de dado da String ASCII:

String é uma série de caracteres e o endereço de memória correspondente, cada caractere é armazenado em um byte. O valor do primeiro byte de uma string é o comprimento da string. Se uma string constante é inserida diretamente no editor de programa ou bloco de dados. A string deve começar e terminar com aspas duplas ("string constant").

Segue abaixo o mapa de memória no formato do tipo de dados da string. O comprimento de uma string deve ter entre 0 e 254 caracteres. O comprimento máximo da string é de 255 bytes.

String length	Character 1	Character 2	Character 3	Character 4	Character 5	Character 254
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	.	

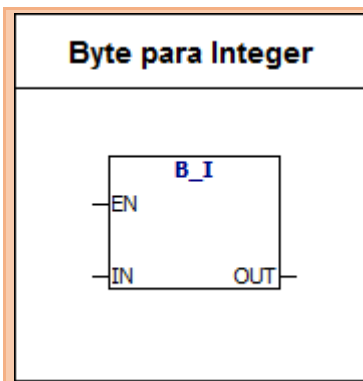
7.5. Conversores

- Conversores
 - B_I
 - I_B
 - I_DI
 - I_S
 - DI_I
 - DI_R
 - DI_S
 - ROUND
 - TRUNC
 - R_S
 - BCD_I
 - I_BCD
 - ITA
 - DTA
 - RTA
 - ATH
 - HTA
 - S_I
 - S_DI
 - S_R
 - DECO
 - ENCO
 - SEG

[Como chamar o arquivo de ajuda](#)

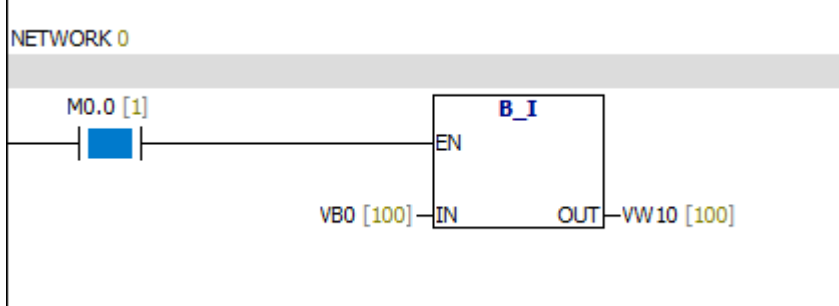
7.5.1. Byte para integer

Entrada / Saída	Operando	Tipo de dado
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *AC, *VD, *LD	Byte
OUT	VW, IW, QW, MW, SW, SMW, LW, AQW, T, C, AC, *VD, *LD, *AC	Integer



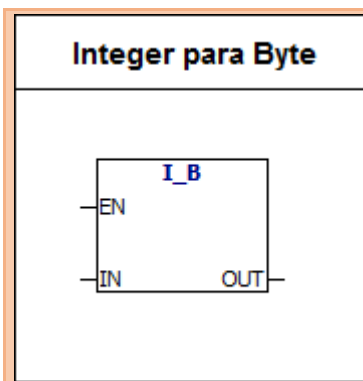
Byte para integer: a instrução B-I converte o valor do byte para o valor inteiro, e o resultado é inserido na variável especificada pelo OUT. Porque o byte não possui um símbolo, o resultado não possui extensão do símbolo.

Exemplo:



7.5.2. Integer para byte

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constant, *VD, *LD, *AC	Integer
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	Byte

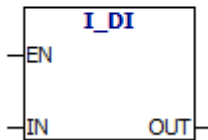


Inteiro para byte: a instrução I-B converte o valor de um inteiro para um valor byte e o resultado é inserido na variável especificada pelo OUT. O intervalo numérico é de 0 a 255. Outros valores resultarão em um estouro, porém a saída não será afetada.

7.5.3. Integer para double integer

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constant, *VD, *LD, *AC	Integer
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Double-integer

Integer para double-integer



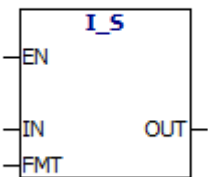
Integer para double integer :

A instrução I-DI converte um valor inteiro em um valor inteiro duplo e o resultado é inserido na variável especificada pelo OUT. O sinal é estendido.

7.5.4. Integer para string

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, constant, AC, *VD, *LD, *AC	Integer
FMT		
OUT	VB, *VD, LB, *AC, *LD	String

Integer para String



Instrução I-S: a instrução converte um valor inteiro em uma string ASCII de 8 caracteres em length. O Formato (FMT) especifica o número de dígitos à direita do ponto decimal. A seqüência de resultados está escrita em 9 bytes consecutivos do OUT. Formato inválido (nnn > 5).

Formato do tipo de dado da String ASCII:

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

Segue abaixo o mapa de memória no formato do tipo de dados da string. O comprimento de uma string pode ter entre 0 e 254 caracteres. O comprimento máximo da string é de 255 bytes.

String length	Character 1	Character 2	Character 3	Character 4	Character 5	Character 254
						.	
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5		

Segue abaixo a definição do número de operação no formato ITS:

```

MSB                                     LSB
7   6   5   4   3   2   1   0
0   0   0   0   c   n   n   n
  
```

C = vírgula (1) ou ponto decimal (0)

NNN = O número de dígitos no lado direito do ponto decimal

O comprimento da String de saída é sempre de 8 caracteres. NNN valores válidos são de 0 a 5. Se nnn = 0, o valor será exibido sem um ponto decimal. Quando o valor de NNN for maior que 5, a saída será exibida como uma String de 8 caracteres do ASCII. C decide usar uma vírgula ou um ponto decimal entre o inteiro e o decimal. Os 4 bits acima da parte superior do formato devem ser zero.

Exemplo:

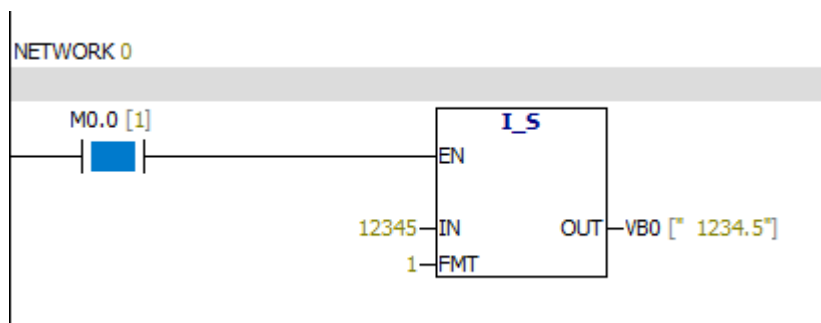
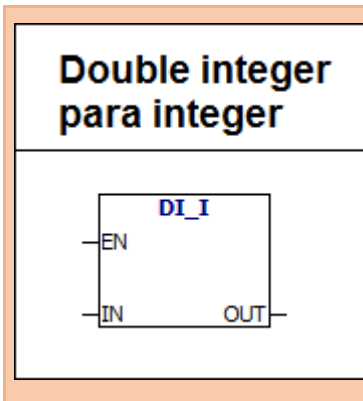


Tabela de monitoramento:

Monitor de Valores								
endereço	Tipo de da...	valor	forçado	endereço	Tipo de da...	valor	forçado	
VW0	INT	354		LB0	SINT	0		
LB30	SINT	2		M0.0	BOOL	0		
VBO	SINT	1						
VB1	SINT	98						
VB2	SINT	0						
VB3	SINT	0						
VB4	SINT	0						

7.5.5. Double integer para integer

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, HC, AC, constant, *VD, *LD, *AC	Double integer
OUT	VW, IW, QW, MW, SW, SMW, LW, AQW, T, C, AC, *VD, *LD, *AC	Integer



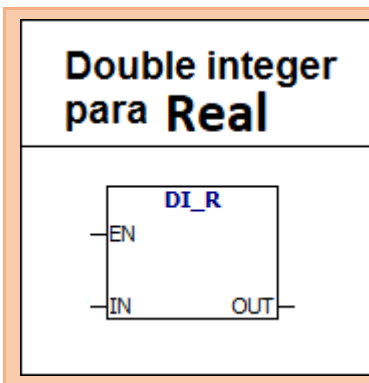
Double integer para integer :

A instrução DI-I converte o valor do inteiro duplo em um valor inteiro e o resultado é inserido na variável especificada pelo OUT.

Valores muito grande irão estourar a variável, porém a saída não será afetada.

7.5.6. Double integer para número real

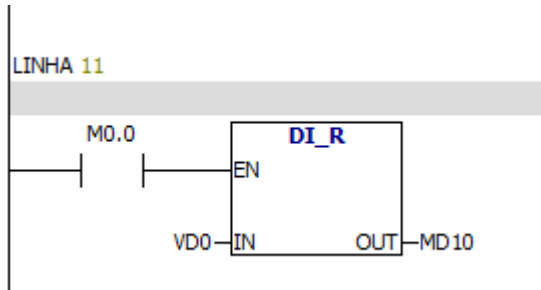
Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, HC, AC, constant, *VD, *AC, *LD	Double integer
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Número real



Double integer para número real :

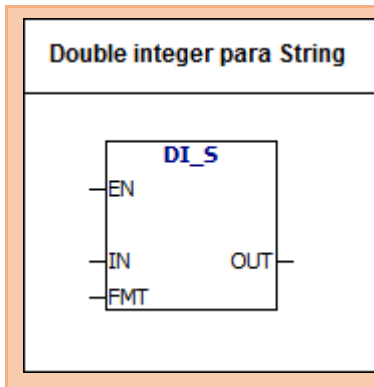
A instrução converte uma variável do tipo Inteiro de 32 bits para uma variável do tipo real de 32 bits e o resultado é inserido na variável especificada pelo OUT.

Exemplo:



7.5.7. Double integer para string

Entrada / Saída	Operando	
IN	VD, ID, QD, MD, SD, SMD, LD, HC, constant, AC, *VD, *AC, *LD	Double integer
FMT	VB, IB, QB, MB, SB, SMB, LB, constant, AC, *VD, *LD, *AC	Byte
OUT	VB, *VD, LB, *AC, *LD	String



Double integer para String:

Instrução DI-S: a instrução converte o inteiro duplo em uma string ASCII de 12 caracteres de tamanho. Formato especifica (FMT), o número de dígitos à direita do ponto decimal. A string de saída é escrita em 13 bytes consecutivos de saída. Formato inválido (nnn > 5).

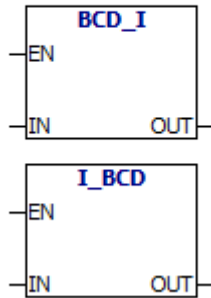
Formato do tipo de dado da String ASCII:

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

Segue abaixo o mapa de memória no formato do tipo de dados da string. O comprimento de uma string pode ter entre 0 e 254 caracteres. O comprimento máximo da string é de 255 bytes.

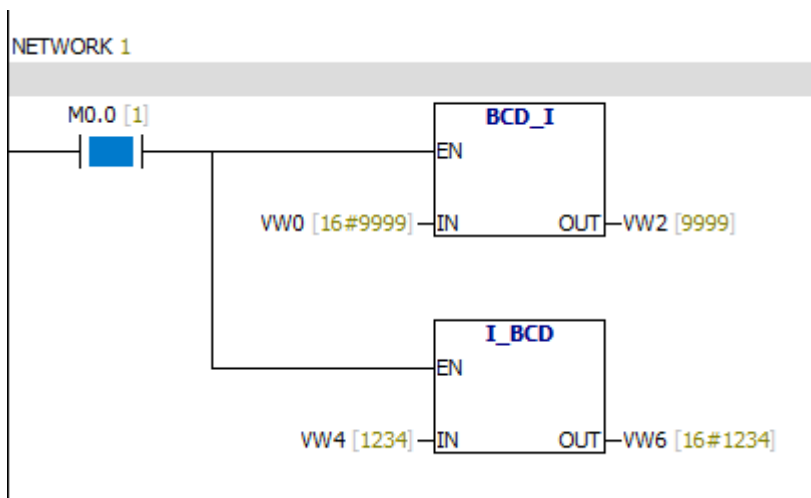
String length	Character 1	Character 2	Character 3	Character 4	Character 5	Character 254
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	.	

Conversão BCD para integer, integer para BCD



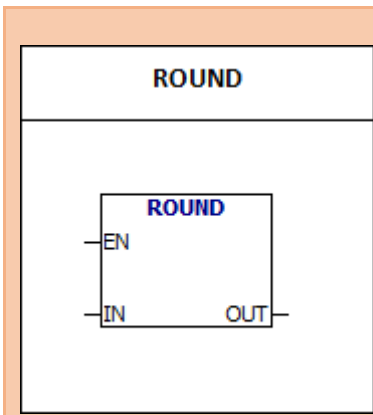
A instrução BCD-I converte o valor decimal codificado em binário para o valor inteiro e carrega o resultado na variável especificada pelo OUT. O intervalo de valores "IN" BCD é 0 a 9999. A instrução Integer para BCD converte o valor inteiro para o valor decimal codificado binário e carrega o resultado na variável especificada pelo OUT. O intervalo de valores de entrada é de 0 a 9999.

Exemplo:



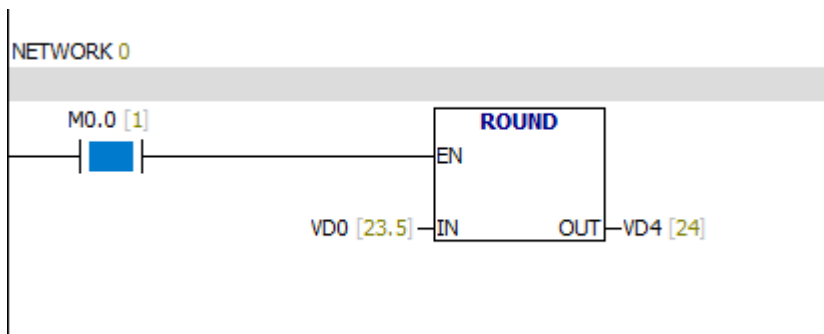
7.5.9. ROUND

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, AC, constant, *VD, *LD, *AC	Número real
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Double integer



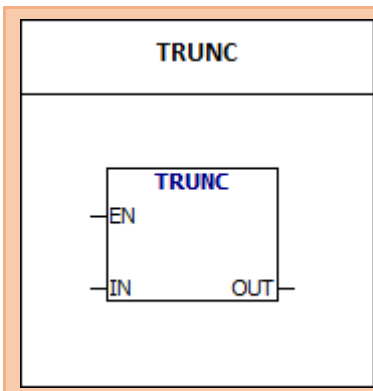
A instrução ROUND converte o valor do número real em um valor inteiro duplo e o resultado é inserido na variável especificada pelo OUT. Se a parte fracionada for igual ou superior a 0,5, a parte inteira será adicionada a 1.

Exemplo:



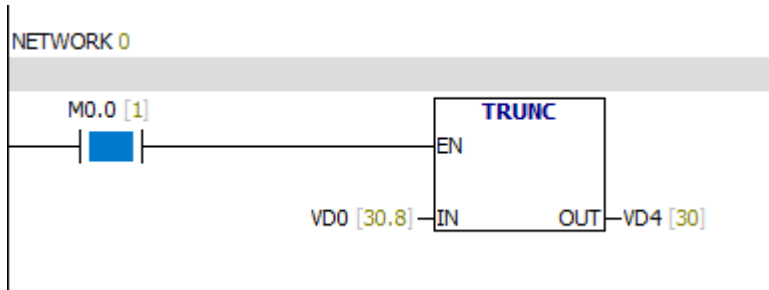
7.5.10. TRUNC

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	Double Integer



TRUNC: A instrução converte 32 bits de número real em inteiro de 32 bits e o resultado é inserido na variável especificada pelo OUT. Somente a parte inteira do número real é convertida e a parte fracionada é descartada.

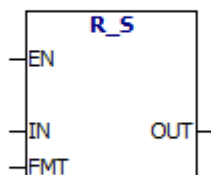
Exemplo:



7.5.11. Número Real para string

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, constant, AC, *VD, *LD, *AC	Número Real
FMT	VB, IB, QB, MB, SB, SMB, LB, constant, AC, *VD, *LD, *AC	Byte
OUT		String

Número Real para String



R-S: A instrução converte o valor do número real em um formato de seqüência ASCII. (FMT) especifica a precisão da conversão à direita do ponto decimal especificado no OUT. O comprimento da string de saída especificado no formato pode ser de 3 a 15 caracteres. O formato dos números reais utilizados no CLP é de no máximo 7 dígitos.

Formato inválido:

- NNN > 5
- SSSS < 3
- SSSS < Quantidade necessária de caracteres

Tipo de dado string constante no formato ASCII :

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

O seguinte mapa de memória mostra o formato do tipo de dados de string. O comprimento de uma string pode ter entre 0 e 254 caracteres. O comprimento máximo da string é de 255 bytes.

String length	Character 1	Character 2	Character 3	Character 4	Character 5	Character 254
						.	
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5		

Segue abaixo a definição da instrução RTS no formato ITS:

MSB	LSB
7 6 5 4 3 2 1 0	s s s s c n n n

SSSS = O comprimento da string de saída

C = vírgula (1) ou ponto decimal (0)

Nnn = O número de caracteres à direita do ponto decimal. O comprimento da seqüência de saída é especificado pelo campo SSSS.0, 1 ou 2 bytes não são válidos. O alcance efetivo do NNN é de 0 a 5. NNN é igual a 0, a saída não mostra nenhum ponto decimal. Quando O valor NNN é maior do que 5 ou quando o comprimento da string de saída especificado é muito pequeno para armazenar o valor da conversão, a seqüência de saída é preenchida com caracteres de espaço ASCII. O bit C especifica usando uma vírgula (C = 1) ou um ponto decimal (C = 0).

Prompt - string de saída de acordo com as seguintes regras:

1. O número positivo é gravado no buffer de saída sem um sinal.
2. O número negativo é gravado no buffer de saída com "-".
3. O zero inicial no lado esquerdo do ponto decimal é comprimido.
4. O tamanho da seqüência de saída deve ser 3 bytes maior do que "nnn"
5. O valor na seqüência de saída deve estar alinhado à direita.

Exemplo :

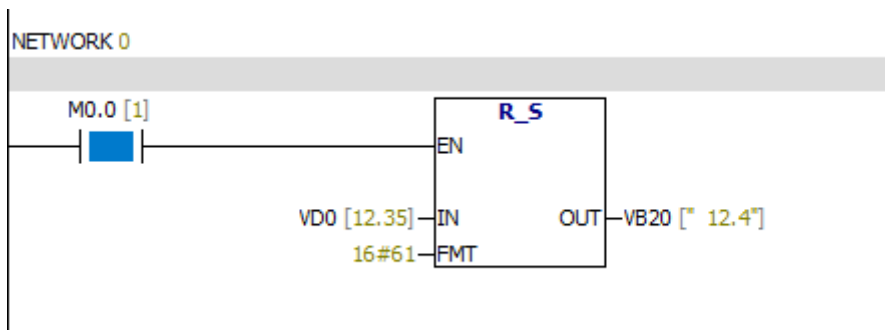


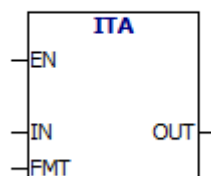
Tabela de monitoramento:

endereço	Tipo de da...	valor	forçado
M0.0	BOOL	1	
VB20	SINT	6	
VB21	SINT	32	
VB22	SINT	32	
VB23	SINT	49	
VB24	SINT	50	
VB25	SINT	46	
VB26	SINT	51	
VB27	SINT	0	
VB28	SINT	0	
VB29	SINT	0	
VB30	SINT	0	

7.5.12. Integer para código ASCII

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constant, *VD, *LD, *AC	Integer
FMT		Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, *VD, *LD, *AC	Byte

Integer para código ASCII

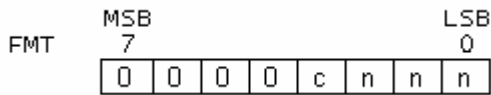


ITA: a instrução converte a palavra inteira em caracteres ASCII. FMT especifica a precisão da conversão à direita do ponto decimal.

O resultado da conversão é colocado nos 8 bytes sucessivos do número do personagem OUT. ASCII é sempre 8 caracteres.

Condição de Erro: nnn > 5

Segue abaixo a definição da instrução ITA no formato FMT:



O tamanho do buffer de saída é sempre de 8 bytes. Nnn = O número de caracteres à direita do ponto decimal. O alcance efetivo do NNN é de 0 a 5. NNN é igual a 0, a saída não mostra um ponto decimal. Quando o valor NNN é maior que 5, a saída A string é preenchida com caracteres de espaço ASCII. O bit C especifica usando uma vírgula (C = 1) ou um ponto decimal (C = 0). Os 4 bits suficientes devem ser 0.

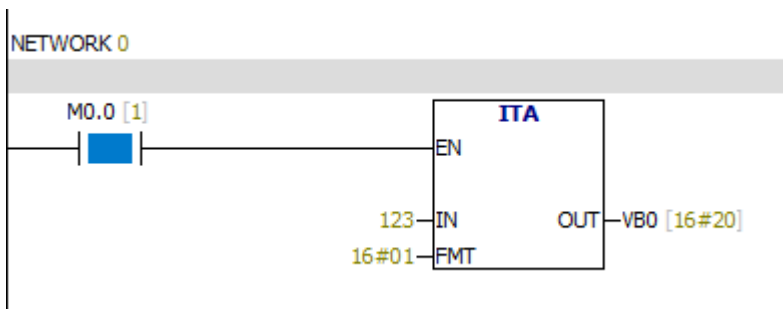
Prompt - string de saída de acordo com as seguintes regras:

1. O número positivo é gravado no buffer de saída sem um sinal.
2. O número negativo é gravado no buffer de saída com "-".
3. O zero inicial no lado esquerdo do ponto decimal é comprimido.
4. O tamanho da seqüência de saída deve ser 3 bytes maior do que "nnn"
5. O valor na seqüência de saída deve estar alinhado à direita.

Exemplo:

	OUT	OUT	OUT	OUT	OUT	OUT	OUT	OUT
	+1	+2	+3	+4	+5	+6	+7	
in = 12			0	.	0	1	2	
in = -123		-	0	.	1	2	3	
in = 1234			1	.	2	3	4	
in = -12345	-	1	2	.	3	4	5	

Exemplo:



Monitor de Valores			
	endereço	Tipo de da...	valor
	M0.0	BOOL	1
	VB0	BYTE	16#20
	VB1	BYTE	16#20
	VB2	BYTE	16#20
	VB3	BYTE	16#20
	VB4	BYTE	16#31
	VB5	BYTE	16#32
	VB6	BYTE	16#2E

Conforme mostrado na figura: A entrada inteira é 123; nnn = 1

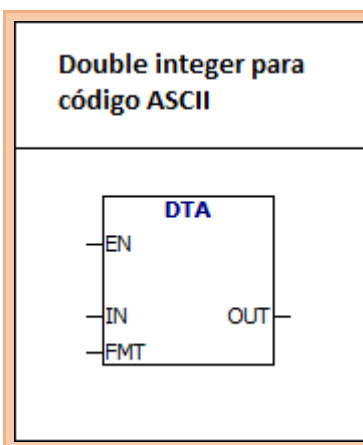
O valor de saída é o seguinte:

```

VB7  16#33  3
VB6  16#2E  .
VB5  16#32  2
VB4  16#31  1
VB3  16#20  Space
VB2  16#20  Space
VB1  16#20  Space
VB0  16#20  Space
    
```

7.5.13. Double integer para o código ASCII

Entrada /Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, HC, constant, AC, *VD, *AC, *LD	Double integer
FMT	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, *VD, *LD, *AC	Byte



DTA: a instrução converte o inteiro duplo em caracteres ASCII. (FMT) especifica a precisão da conversão à direita do ponto decimal. O resultado da conversão é colocado nos 12 bytes sucessivos do OUT.

Condição de Erro:

FMT alto valor de 4 bits maior que 0

NNN > 5

Segue abaixo a definição da instrução DTA no formato FMT:



O tamanho do buffer de saída é sempre 12 bytes. NNN = O número de caracteres à direita do ponto decimal. O intervalo efetivo do NNN é de 0 a 5. NNN é igual a 0, a saída não mostra nenhum ponto decimal. Quando o valor NNN é superior a 5, a sequência de saída é preenchida com caracteres de espaço ASCII. O bit C especifica usando uma vírgula (C = 1) ou um ponto decimal (C = 0). Os 4 bits maiores devem ser 0.

Prompt - string de saída de acordo com as seguintes regras:

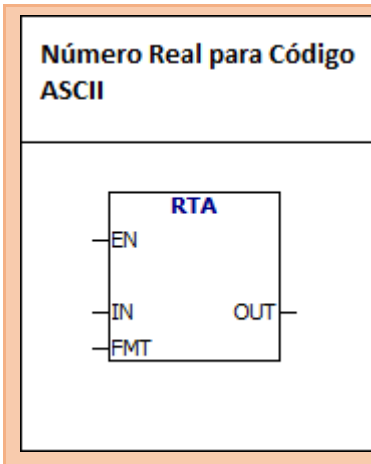
1. O número positivo é gravado no buffer de saída sem um sinal.
2. O número negativo é gravado no buffer de saída com "-".
3. O zero inicial no lado esquerdo do ponto decimal é comprimido.
4. O tamanho da seqüência de saída deve ser 3 bytes maior do que "nnn"
5. O valor na seqüência de saída deve estar alinhado à direita.

Exemplo:

OUT character	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11
in = -12						-	0	.	0	0	1	2
in = 1234567				1	2	3	.	4	5	6	7	

7.5.14. Número Real para código ASCII

Entrada/Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, AC, constant, *VD, *LD, *AC	Número Real
FMT	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, *VD, *LD, *AC	Byte



RTA: a instrução converte o número real em caracteres ASCII. (FMT) especifica a precisão da conversão à direita do ponto decimal. O resultado da conversão é colocado no buffer de saída do OUT. O comprimento do buffer de saída é de 3 a 15 caracteres.

Condições de Erro:

nnn > 5

ssss < 3

ssss < Número de caracteres em OUT.

Segue abaixo a definição da instrução RTA no formato FMT:



O comprimento da saída é especificado pelo campo SSSS.0, 1 ou 2 bytes não são válidos. O alcance efetivo do NNN é de 0 a 5. NNN é igual a 0, a saída não mostra nenhum ponto decimal. Quando O valor NNN é maior do que 5 ou quando o comprimento da string de saída especificado é muito pequeno para armazenar o valor da conversão, a seqüência de saída é preenchida com caracteres de espaço ASCII. O bit C especifica usando uma vírgula (C = 1) ou um ponto decimal (C = 0).

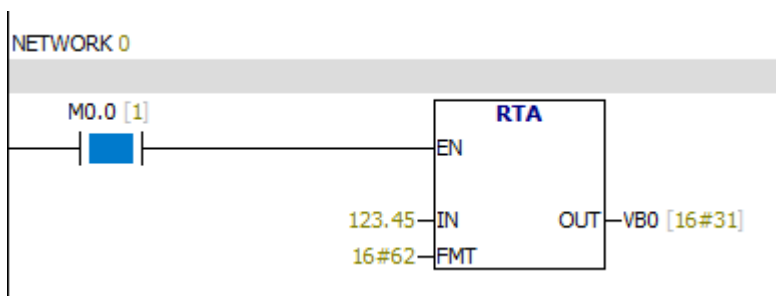
A saída esta de acordo com as seguintes regras:

1. O número positivo é gravado no buffer de saída sem um sinal.
2. O número negativo é gravado no buffer de saída com "-".
3. O zero inicial no lado esquerdo do ponto decimal é descartado.
4. O número de caracteres à direita do ponto decimal é igual ao valor de "nnn".
5. O tamanho da seqüência de saída deve ser 3 bytes maior que "nnn".
6. O valor na string de saída deve estar alinhado à direita.

Exemplo:

	OUT	OUT	OUT	OUT	OUT	OUT
		+1	+2	+3	+4	+5
in = 1234.5	1	2	3	4	.	5
in = -0.0004				0	.	0
in = -3.67526			-	3	.	7
in = 1.95				2	.	0

Exemplo:



Monitor de Valores			
	endereço	Tipo de da...	valor
	M0.0	BOOL	1
	VB0	BYTE	16#20
	VB1	BYTE	16#20
	VB2	BYTE	16#20
	VB3	BYTE	16#20
	VB4	BYTE	16#31
	VB5	BYTE	16#32
	VB6	BYTE	16#2E
	VB7	BYTE	16#33
	VB8	BYTE	16#00
	VB9	BYTE	16#00
	VB10	BYTE	16#00

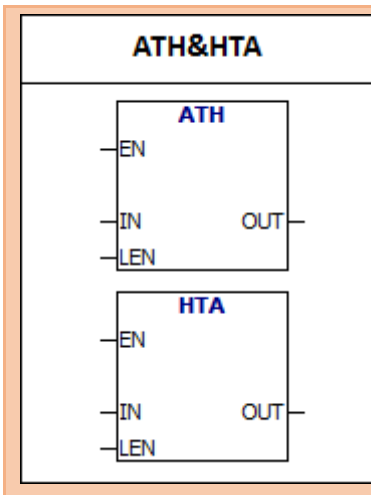
Converte o número real 123.45 em código ASCII. A saída é de 6 bytes.

OUT:

VB0	VB1	VB2	VB3	VB4	VB5
16#31	16#32	16#33	16#2E	16#34	16#35
1	2	3	.	4	5

7.5.15. ATH&HTA

Entrada /output	Operando	Tipo de dado
IN, OUT	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD	Byte
LEN	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte



ASCII para HEX a instrução converte os caracteres ASCII começando com "IN" para hexadecimal começando em "out". O comprimento máximo da string ASCII é de 255 caracteres.

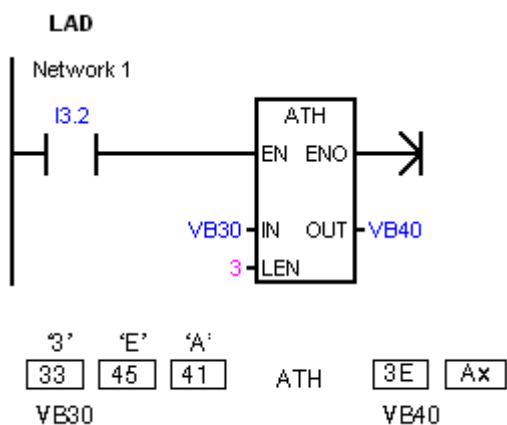
HEX to ASCII a instrução converte os dígitos em hexadecimal começando em "IN" para caracteres ASCII começando em "out".

Caractere de entrada ASCII válido: números de 0 a 9 e letras maiúsculas A a F.

Códigos ASCII: 30 a 39 e 41 a 46.

Condição de Erro: código ASCII ilegal

Exemplo:



Exemplo:

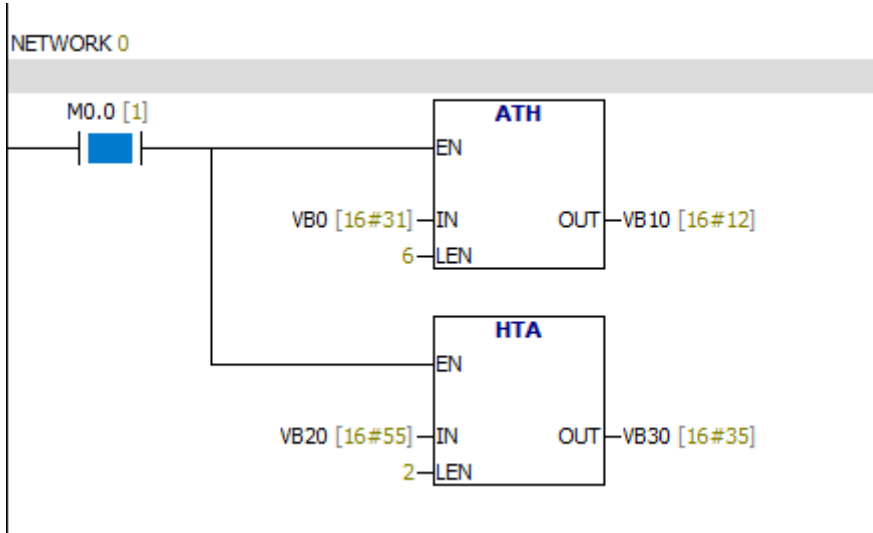


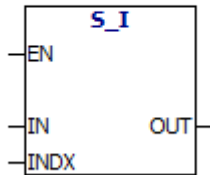
Tabela de monitoramento:

Status Chart			
Address	Data Type	Value	
M0.0	BOOL	1	
VB0	BYTE	16#31	
VB1	BYTE	16#32	
VB2	BYTE	16#33	
VB3	BYTE	16#2E	
VB4	BYTE	16#34	
VB5	BYTE	16#35	

7.5.16. String para integer

Entrada / Saída	Operando	Tipo de dado
IN	VB, constant string, LB, *VD, *LD, *AC	String
INDX	VB, IB, QB, MB, SB, SMB, LB, constant, AC, *VD, *LD, *AC	
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, AC, *VD, *LD, *AC	Integer

String para integer



S-I: a instrução converte o valor da string "IN" para o valor inteiro armazenado no OUT, começando com a localização offset INDX.

Formato do tipo de dado da String ASCII:

String é uma série de caracteres, cada caracter é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

O seguinte mapa de memória mostra o formato do tipo de dados de string. O comprimento de uma string pode ter entre 0 e 254 caracteres. O comprimento máximo da string é de 255 bytes.

String length	Character 1	Character 2	Character 3	Character 4	Character 5	Character 254
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5		

O valor INDX normalmente é definido como 1, começando a conversão a partir do primeiro caracter do valor string. INDX pode ser configurado para outros valores. Esse método pode ser usado quando a string de entrada contém caracteres que não são necessários para serem convertidos. Por exemplo, se a string de entrada é "Temperatura: 77.8", você pode definir o valor INDX 13 para ignorar os caracteres "Temperatura:". Quando o fim da string é atingido ou quando o primeiro caractere inválido é encontrado, a conversão é encerrada. O caractere inválido é qualquer caractere diferente do número (0-9). A tabela a seguir mostra exemplos de cordas Entradas enteras válidas e inválidas:

A tabela a seguir mostra exemplos de string de entrada inteira válidas e inválidas:

Entrada válida de String, String para integer/Double integer

Input String	Output Integer
"123"	123
"-00456"	-456
"123.45"	123
"+2345"	2345
"000000123ABCD"	123

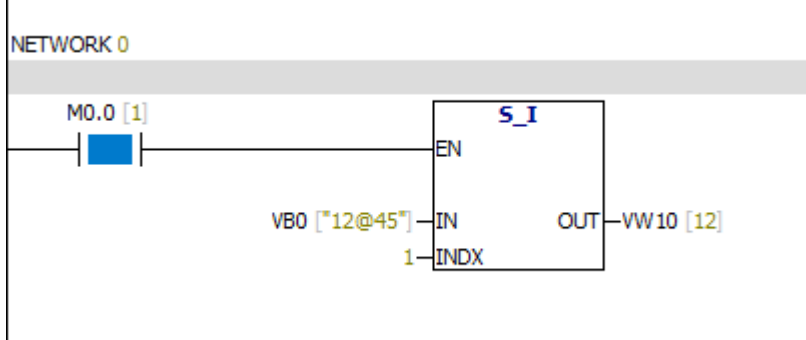
Entrada válida de String, String para Real

Input String	Output Real
"123"	123.0
"-00456"	-456.0
"123.45"	123.45
"+2345"	2345.0
"000000123"	.000000123

Entrada inválida de String

Input String
"A123"
" "
"++123"
"+-123"
"+ 123"

Exemplo:

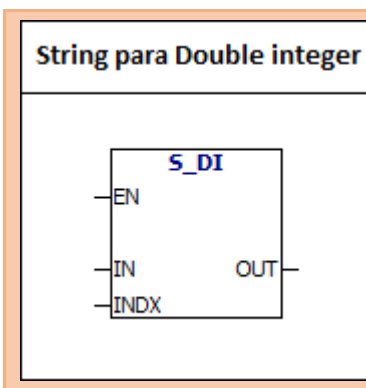


Monitor de Valores			
	endereço	Tipo de da...	valor
	M0.0	BOOL	1
	VB0	BYTE	16#20
	VB1	BYTE	16#20
	VB2	BYTE	16#20
	VB3	BYTE	16#20
	VB4	BYTE	16#31
	VB5	BYTE	16#32
	VB6	BYTE	16#2E

Digite o valor "12@45" na string. A instrução S-I converte a string do primeiro caracter e o resultado é um número inteiro 12.

7.5.17. String para double integer

Entrada / Saída	Operando	Tipo de dado
IN	VB, constant string, LB, *VD, *LD, *AC	String
INDX	VB, IB, QB, MB, SB, SMB, LB, constant, AC, *VD, *LD, *AC	Byte
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Double integer



S-DI: a instrução converte o valor da string "IN" para o valor inteiro duplo armazenado no "OUT", começando com a localização offset INDX.

Formato do tipo de dado da String ASCII:

String é uma série de caracteres, cada caracter é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

O seguinte mapa de memória mostra o formato do tipo de dados de string. O comprimento de uma string pode ter entre 0 e 254 caracteres. O comprimento máximo da string é de 255 bytes.

String length	Character 1	Character 2	Character 3	Character 4	Character 5	Character 254
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	.	

O valor INDX normalmente é definido como 1, começando a conversão a partir do primeiro caractere do valor string. INDX pode ser configurado para outros valores. Esse método pode ser usado quando a seqüência de entrada contém caracteres que não são necessários para serem convertidos.

Por exemplo, se a seqüência de entrada for "Temperatura: 77.8", você pode definir o valor INDX 13 para ignorar os caracteres "Temperatura:". Quando o fim da string é atingido ou quando o primeiro caractere inválido é encontrado, a conversão é encerrada. O caractere inválido é qualquer caractere diferente do número (0-9).

A tabela a seguir mostra exemplos de string de entrada inteira válidas e inválidas:

Entrada válida de String, String para integer/Double integer

Input String	Output Integer
"123"	123
"-00456"	-456
"123.45"	123
"+2345"	2345
"000000123ABCD"	123

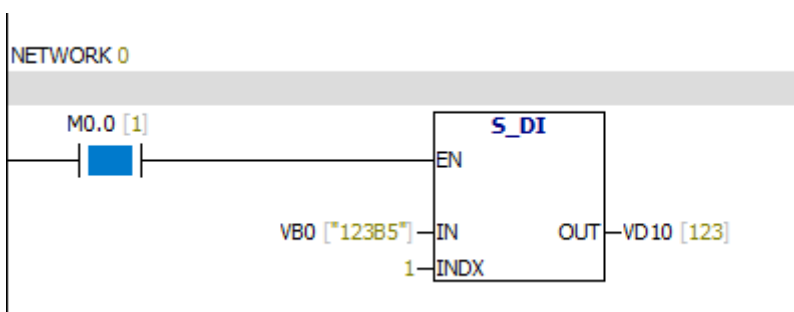
Entrada válida de String, String para Real

Input String	Output Real
"123"	123.0
"-00456"	-456.0
"123.45"	123.45
"+2345"	2345.0
"000000123"	.000000123

Entrada inválida de String

Input String
"A123"
" "
"++123"
"+-123"
"+ 123"

Exemplo:



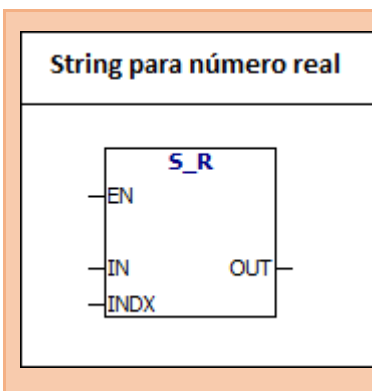
Monitor de Valores			
	endereço	Tipo de da...	valor
	M0.0	BOOL	1
	VB0	BYTE	16#20
	VB1	BYTE	16#20
	VB2	BYTE	16#20
	VB3	BYTE	16#20
	VB4	BYTE	16#31
	VB5	BYTE	16#32
	VB6	BYTE	16#2E

Digite o valor "123B5" na string. A instrução S-DI converte a string do primeiro caractere e o resultado é um número inteiro 123.

Como B é um caractere "especial", os caracteres após B não são mais convertidos.

7.5.18. String para número real

Entrada / Saída	Operando	Tipo de dado
IN	VB, constant string, LB, *VD, *LD, *AC	String
INDX	VB, IB, QB, MB, SB, SMB, LB, Constant, AC, *VD, *LD, *AC	Byte
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Número Real



S-R: A instrução converte o valor da string "IN" para o valor do número real armazenado no "OUT", começando com offset de localização INDX.

Formato do tipo de dado da String ASCII:

String é uma série de caracteres, cada caracter é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

O seguinte mapa de memória mostra o formato do tipo de dados de string. O comprimento de uma string pode ter entre 0 e 254 caracteres. O comprimento máximo da string é de 255 bytes.

String length	Character 1	Character 2	Character 3	Character 4	Character 5	Character 254
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	.	

O valor INDX normalmente é definido como 1, começando a conversão a partir do primeiro caractere do valor string. INDX pode ser configurado para outros valores. Esse método pode ser usado quando a seqüência de entrada contém caracteres que não são necessários para serem convertidos.

Por exemplo, se a seqüência de entrada for "Temperatura: 77.8", você pode definir o valor INDX 13 para ignorar os caracteres "Temperatura:". Quando o fim da string é atingido ou quando o primeiro caractere inválido é encontrado, a conversão é encerrada. O caractere inválido é qualquer caractere diferente do número (0-9).

A tabela a seguir mostra exemplos de string de entrada inteira válidas e inválidas:

Entrada válida de String, String para integer/Double integer

Input String	Output Integer
"123"	123
"-00456"	-456
"123.45"	123
"+2345"	2345
"000000123ABCD"	123

Entrada válida de String, String para Real

Input String	Output Real
"123"	123.0
"-00456"	-456.0
"123.45"	123.45
"+2345"	2345.0
"000000123"	.000000123

Entrada inválida de String

Input String
"A123"
" "
"++123"
"+-123"
"+ 123"

Exemplo:

NETWORK 0

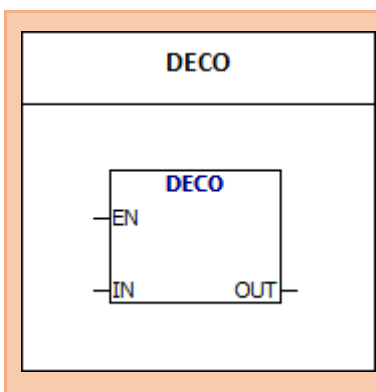
Monitor de Valores

endereço	Tipo de da...	valor
M0.0	BOOL	1
VB0	BYTE	16#20
VB1	BYTE	16#20
VB2	BYTE	16#20
VB3	BYTE	16#20
VB4	BYTE	16#31
VB5	BYTE	16#32
VB6	BYTE	16#2E

String de entrada "2.345" e saída do número real 2.345

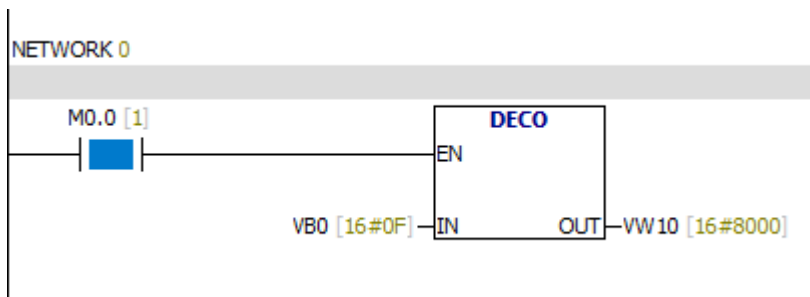
7.5.19. DECO

Entrada / Saída	Operando	Tipo de dado
Entrada	VB, IB, QB, MB, SMB, LB, SB, AC, constant, *VD, *LD, *AC	Byte
OUT	VW, IW, QW, MW, SMW, LW, SW, AQW, T, C, AC, *VD, *AC, *LD	Word



O valor baixo de quatro bits do byte de entrada é N, o número da palavra de saída é igual a 1. Os outros bits da palavra de saída são definidos como 0.

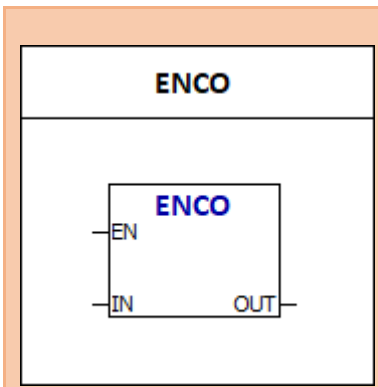
Exemplo:



O valor baixo de quatro bits de VBO é 15, o 15º bit do VW10 é igual a 1. Os outros bits do VW10 estão configurados para 0.

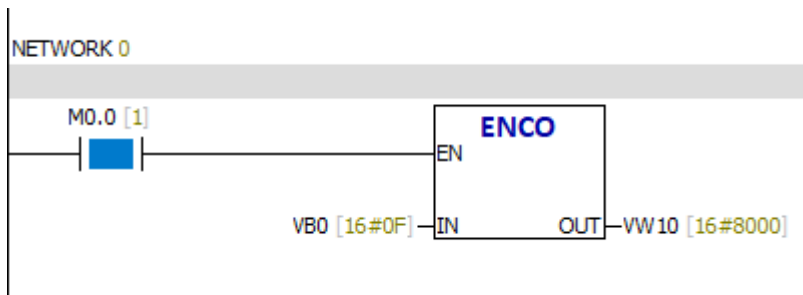
7.5.20. ENCO

Entrada / Saída	Operando	Data type
Entrada	VW, IW, QW, MW, SMW, LW, SW, AIW, T, C, AC, constant, *VD, *AC, *LD	Word
OUT	VB, IB, QB, MB, SMB, LB, SB, AC, *VD, *LD, *AC	Byte



ENCO: O n-ésimo bit da palavra de entrada é igual a 1. O valor baixo de quatro bits do byte de saída é n.

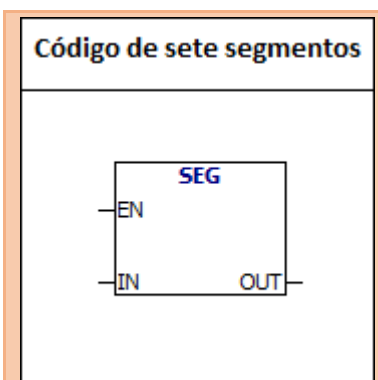
Exemplo:



Como mostrado na figura acima: O 15º bit da palavra de entrada vw0 é igual a 1. O valor baixo de quatro bits do byte de saída vb10 é 15.

7.5.21. Código de sete segmentos

Entrada /output	Operando	Tipo de dado
Entrada	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *AC, *LD	Byte
OUT	VB, IB, QB, MB, SMB, LB, AC, *VD, *AC, SB, *LD	Byte



SEG: as instruções geram os bits de sete segmentos.

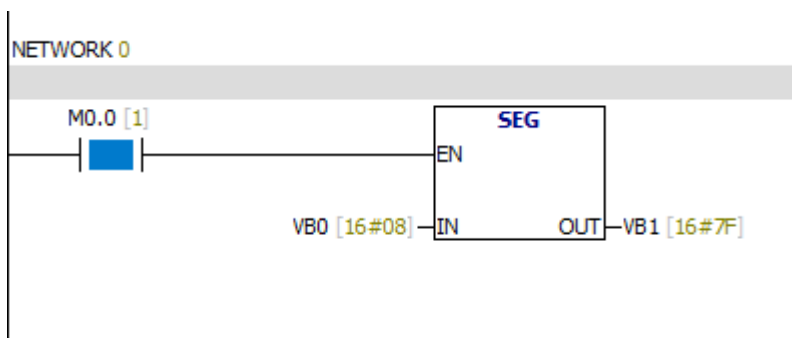
O valor baixo de quatro bits do byte de entrada é convertido.

Tabela de código de sete segmentos:

(IN) LSD	Segment Display	(OUT) - g f e d c b a
0		0 0 1 1 1 1 1 1
1		0 0 0 0 0 1 1 0
2		0 1 0 1 1 0 1 1
3		0 1 0 0 1 1 1 1
4		0 1 1 0 0 1 1 0
5		0 1 1 0 1 1 0 1
6		0 1 1 1 1 1 0 1
7		0 0 0 0 0 1 1 1

(IN) LSD	Segment Display	(OUT) - g f e d c b a
8		0 1 1 1 1 1 1 1
9		0 1 1 0 0 1 1 1
A		0 1 1 1 0 1 1 1
B		0 1 1 1 1 1 0 0
C		0 0 1 1 1 0 0 1
D		0 1 0 1 1 1 1 0
E		0 1 1 1 1 0 0 1
F		0 1 1 1 0 0 0 1

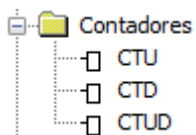
Exemplo:



Análise:

O valor baixo de quatro bits de VB0 é 8. O valor do byte de saída VB1 é 16 # 7F. O resultado da conversão de VB1 para binário é 2 # 0111 1111.

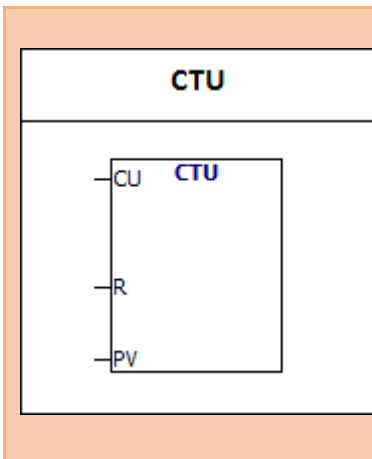
7.6. Contadores



[Como chamar o arquivo de ajuda](#)

7.6.1. CTU

Entrada /output	Operando	Tipo de dado
C xxx	Constant (C0—C255)	Word
CU (LAD)	Enable bit	Booleana
CU (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
R (LAD)	Enable bit	Booleana
R (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
PV	VW, IW, QW, MW, SMW, LW, AIW, AC, T, C, constant, *VD, *AC, *LD, SW	Integer

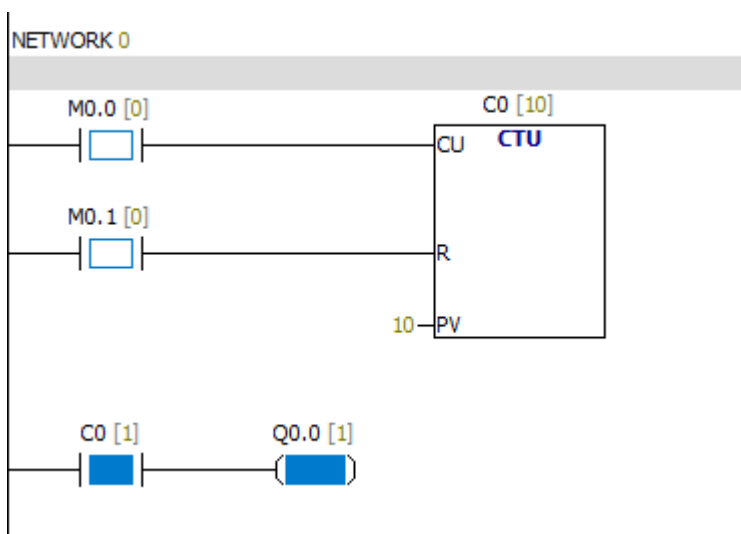


O bit CU obtém um nível alto e o valor atual do contador mais 1. Quando o valor atual é maior ou igual ao valor pré-definido, o bit contador é aberto. Quando R obtém um nível alto, o contador é restaurado. O valor máximo do contador é 32767.

Escala de contador: C xxx = C0 ~ C255

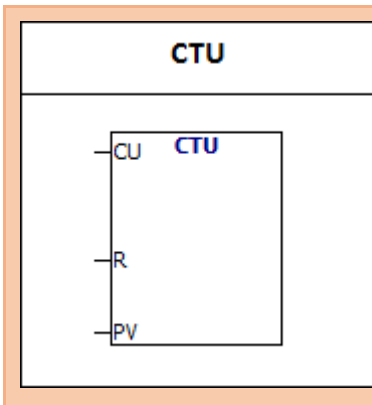
O número contador de cada contador é diferente.

Exemplo:



7.6.2. CTD

Entrada / Saída	Operando	Tipo de dado
Cxxx	Constant (C0—C255)	Word
CD (LAD)	Enable bit	Booleana
CD (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
LD (LAD)	Enable bit	Booleana
LD (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
PV	VW, IW, QW, MW, LW, SMW, AC, T, C, AIW, constant, *VD, *AC, *LD, SW	Integer

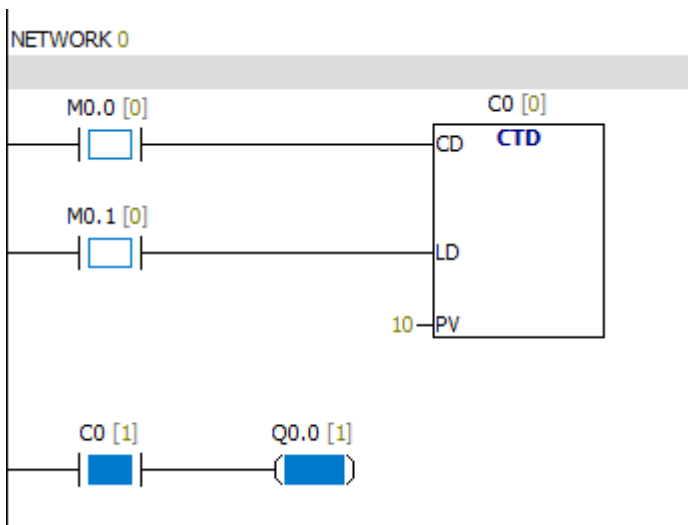


O bit CD é convertido de 0 a 1, valor atual menos 1. Quando o valor atual é igual a 0, o contador é aberto e o contador para de contar. Quando o bit LD é igual a 1, o bit contador é reiniciado e o valor predefinido é carregado no valor atual.

Escala de contador: C xxx = C0 ~ C255

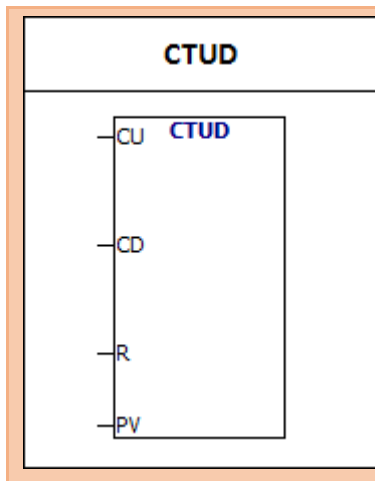
Atenção: O número de cada contador é diferente.

Exemplo:



7.6.3. CTUD

Entrada / Saída	Operando	Tipo de dado
C xxx	Constant (C0—C255)	Word
CU, CD (LAD)	Enable bit	Booleana
CU, CD (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
R (LAD)	Enable bit	Booleana
R (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
PV	VW, IW, QW, MW, LW, SMW, AC, T, C, AIW, constant, *VD, *AC, *LD, SW	Integer

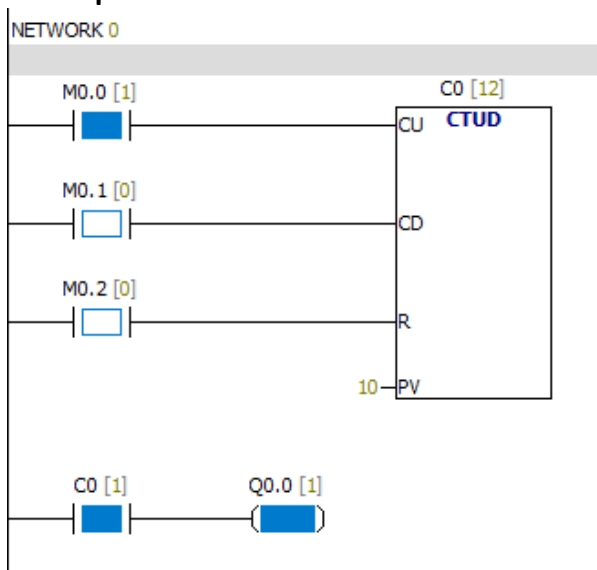


Quando o bit CU obtém um nível alto o contador incrementa 1 ao valor atual. Quando o bit CD é convertido de 0 para 1 eo valor atual menos 1. Quando o valor atual for maior ou igual ao valor predefinido, o bit contador Abre. O valor máximo do contador é 32767 eo valor mínimo é -32768. Quando R obtém um nível alto, o contador é restaurado.

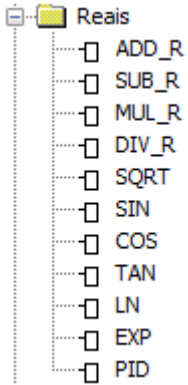
Escala de contador: C xxx = C0 ~ C255

Atenção: O número contador de cada contador é diferente.

Exemplo:



7.7. Real



7.7.1. ADD-R&SUB-R

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VD, ID, QD, MD, SD, SMD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Número Real

ADD-R&SUB-R	
	ADD-R: Adicionando N1 e N2, o resultado é colocado no buffer de saída.
	SUB-R: N1 menos N2, o resultado é colocado no buffer de saída.
	N1, N2 e OUT são 32 bits de números reais.
	Em LAD e FBD: $IN1 + IN2 = OUT$
	$IN1 - IN2 = OUT$

Bit memória especial:

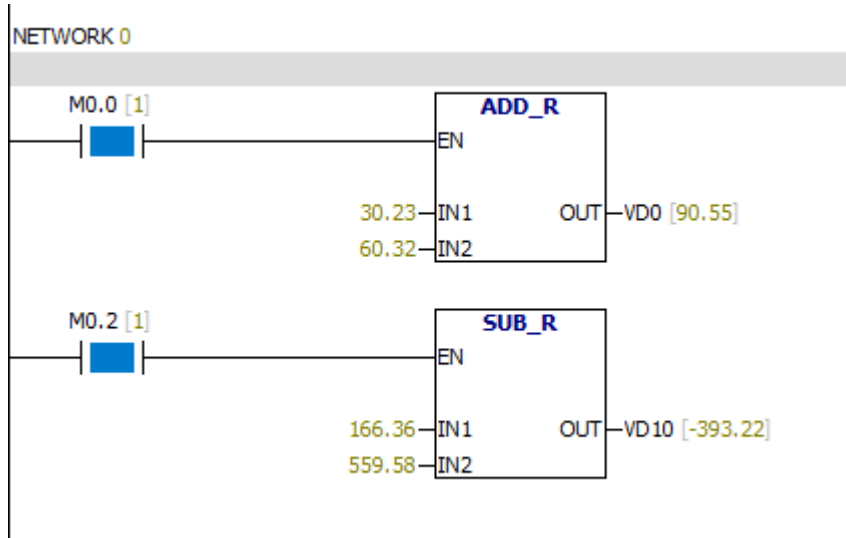
SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

SM1.1 É usado para indicar erros de estouro e valores inválidos.

Exemplo:



7.7.2. MUL - R&DIV - R

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VD, ID, QD, MD, SD, SMD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Número Real

MUL - R&DIV - R	
	IN1 multiplicado por IN2, o resultado é colocado no buffer de saída.
	DIV - R: IN1 dividido por IN2, o resultado é colocado no buffer de saída.
	IN1, IN2 e OUT são 32 bits de números reais.
	Em LAD e FBD: $IN1 * IN2 = OUT$
	$IN1 / IN2 = OUT$

Condições de Erro:

SM1.1 Estouro

SM1.3 Divisão por 0

Bit memória especial:

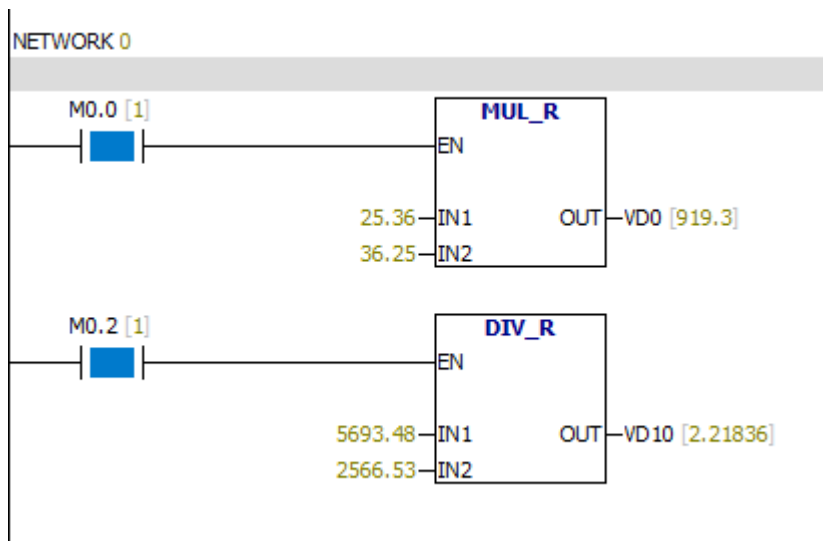
SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

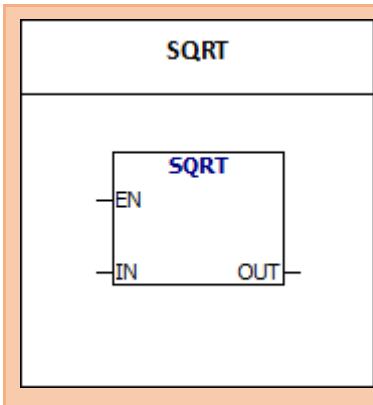
SM1.3 Divisão por 0

Exemplo:



7.7.3. SQRT

Entrada /Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Número Real



SQRT: Digite um número real de 32 bits (IN). Tome "IN" valor da raiz quadrada e "OUT" o resultado em 32 bits em um número real.

Fórmula:

$$\sqrt{IN} = OUT$$

Condições de Erro:

SM1.1 Estouro

Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

SM1.1 É usado para indicar erros de estouro e valores inválidos.

Exemplo:

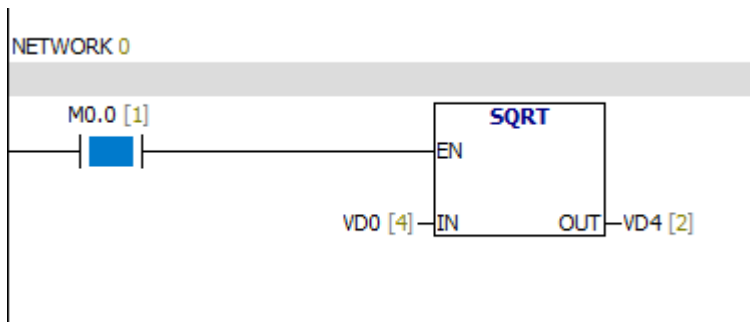
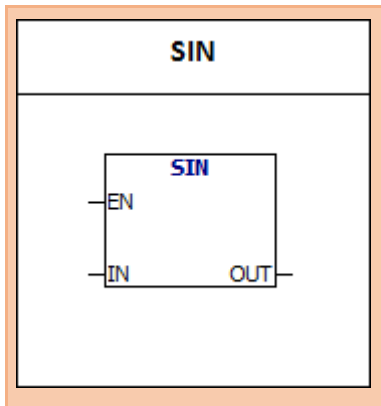


Tabela de Monitoramento:

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
M0.0	BOOL	1					
VD0	REAL	4					
VD4	REAL	2					

7.7.4. SIN

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Número Real



SIN: Executa operações trigonométricas, valor de entrada em radiano e coloque o resultado em OUT. Você pode usar o valor em graus multiplicando por 1.745329E-2 para converter o valor do radiano. O valor da entrada "IN" é radiano.

SM1.1 é usado para indicar erros de estouro e valores inválidos.

Condições de Erro:

SM1.1 Estouro

Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

Exemplo:

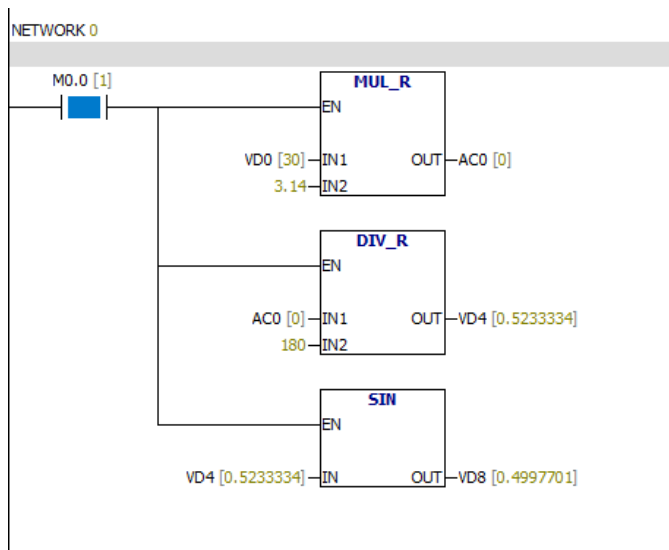
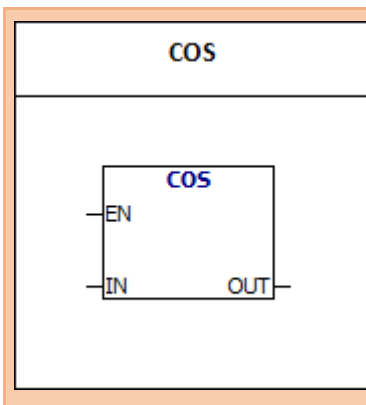


Tabela de Monitoramento:

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
M0.0	BOOL	1					
V00	REAL	30					

7.7.5. COS

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Número Real



COS: Executa operações trigonométricas, valor de entrada em radiano e coloque o resultado em OUT. Você pode usar o valor em graus multiplicando por 1.745329E-2 para converter o valor do radiano. O valor da entrada "IN" é radiano.

SM1.1 é usado para indicar erros de estouro e valores inválidos.

Condições de Erro:

SM1.1 Estouro

Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

Exemplo:

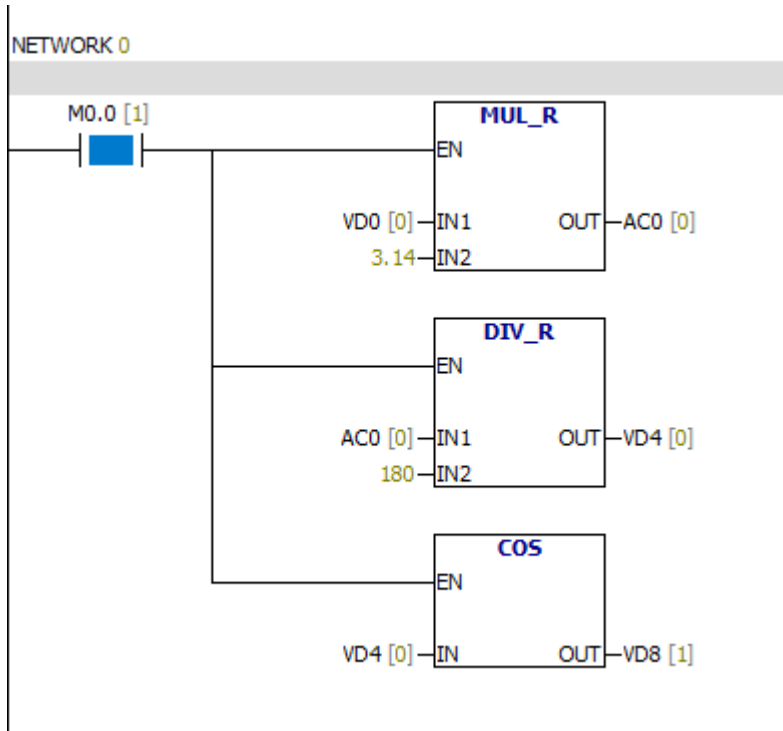
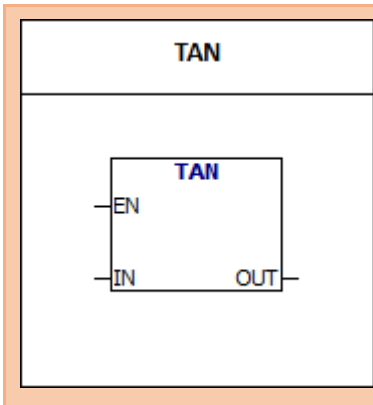


Tabela de monitoramento:

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
M0.0	BOOL	1					
VD0	REAL	0					

7.7.6. TAN

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Número Real



TAN: Executa operações trigonométricas, valor de entrada em radiano e coloque o resultado em OUT. Você pode usar o valor em graus multiplicando por 1.745329E-2 para converter o valor do radiano. O valor da entrada "IN" é radiano.

SM1.1 é usado para indicar erros de estouro e valores inválidos.

Condições de Erro:

SM1.1 Estouro

SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

Exemplo:

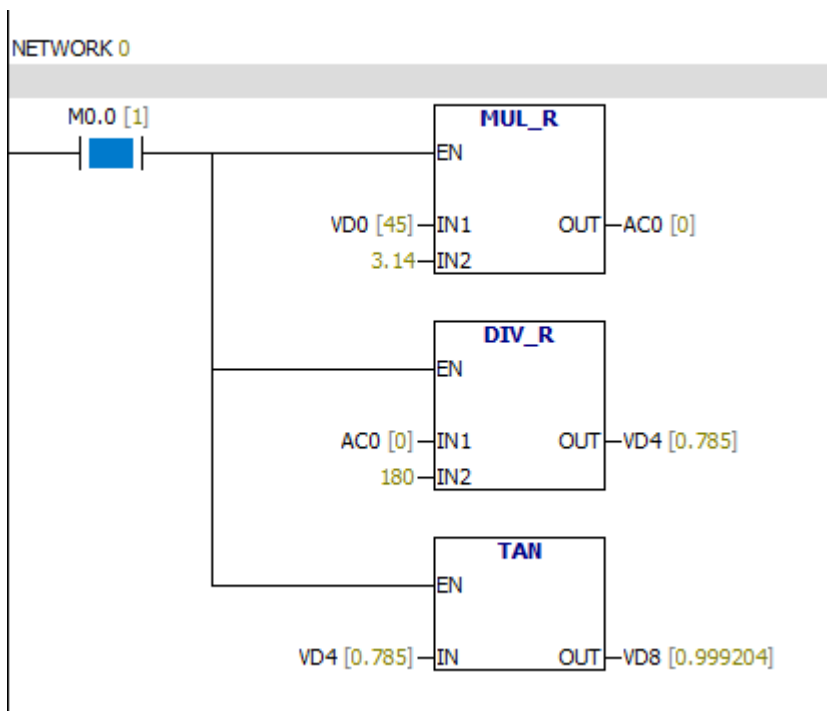
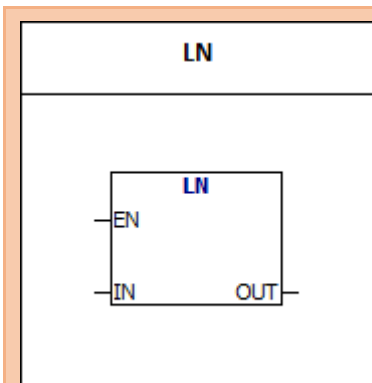


Tabela de monitoramento:

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
MD0	BOOL	1					
VD0	REAL	45					
VD8	REAL	0.999204					

7.7.7. LN

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Número Real



LN: Use o valor de entrada para executar o cálculo do logaritmo e coloque o resultado em OUT.

O valor de saída / 2.302585 \approx Logaritmo de 10

SM1.1 é usado para indicar erros de estouro e valores inválidos.

Condições de Erro:

SM1.1 Estouro

Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

Exemplo:

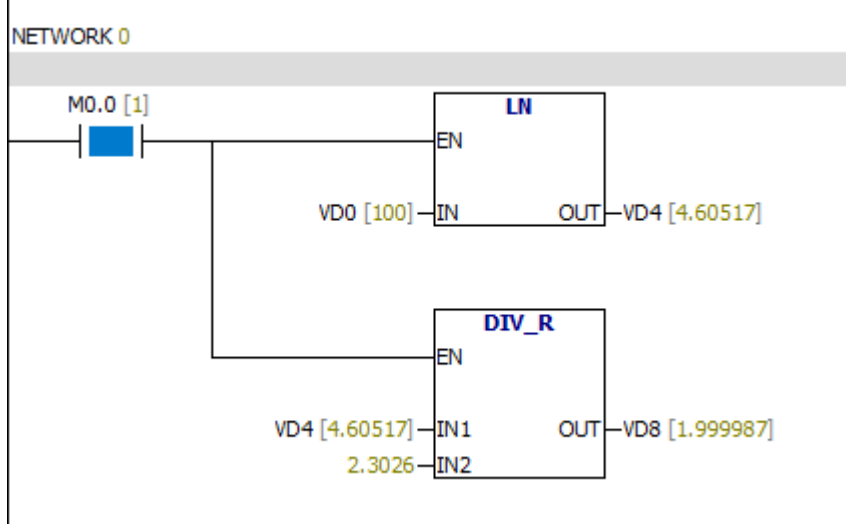
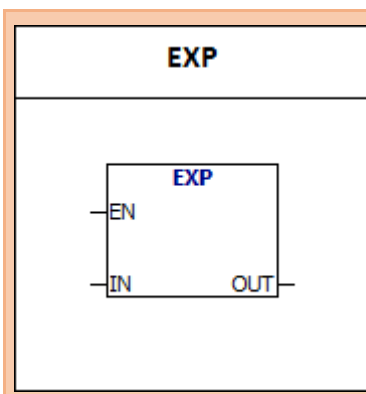


Tabela de monitoramento:

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
M0.0	BOOL	1					
VD0	REAL	100					

7.7.8. EXP

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SMD, SD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Número Real



EXP: O valor de entrada é N e o valor de saída é e^N .

N é um número real.

SM1.1 é usado para indicar erros de estouro e valores inválidos.

Exemplo :

5 ao cubo = $5^3 = \text{EXP}(3 * \text{LN}(5)) = 125$

The cube root of 125 = $125^{(1/3)} = \text{EXP}(1/3 * \text{LN}(125)) = 5$

5 raiz cúbica de 125 = $5^{(3/2)} = \text{EXP}(3/2 * \text{LN}(5)) = 11.18034$

Condições de Erro:

0006 Endereço Indireto

SM1.1 Estouro

Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

Exemplo:

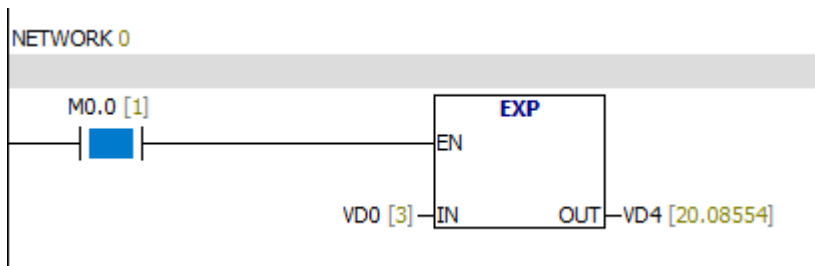
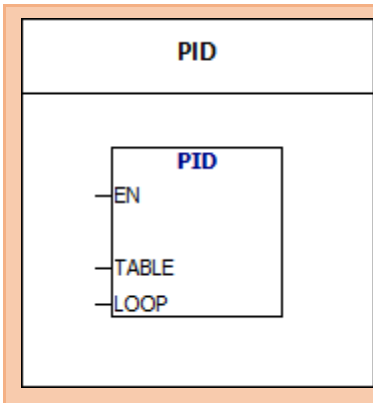


Tabela de monitoriamento:

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
M0.0	BOOL	1					
VDO	REAL	3					

7.7.9. PID

Entrada / Saída	Operando	Tipo de dado
TBL	VB	Byte
LOOP	Constant (0 to 7)	Byte



De acordo com os parâmetros no TBL, a instrução PID executa a operação PID. Após 8 instruções PID podem ser usadas no programa. O PID não pode ser o mesmo, caso contrário pode causar uma interferência. Os parâmetros TBL incluem: Processo, valor definido, saída, ganho, tempo de amostragem, tempo de integração, tempo diferencial, ultimo termo da integral, a última quantidade de processo.

A tabela de parâmetros contém 36 bytes:

Offset	Meaning	Format	Type	Explain
0	PV_n Process quantity	DINT	Input	0.0~1.0
4	SP_n Set point	DINT	Input	0.0~1.0
8	Mn Output value	DINT	Input/Output	0.0~1.0
12	Kc gain		Input	Ratio constant
16	Ts Sampling time	DINT	Input	Ms Positive
20	T_I Integral time	DINT	Input	S Positive
24	T_D Differential time	DINT	Input	S Positive
28	MI_{n-1} Last time integral value	DINT	Input/Output	Last time integral value
32	PV_{n-1} Last time process	DINT	Input/Output	Last time process

Fórmula matemática da instrução PID:

$$Mn = MPn + MInn$$

Mn : valor de saída

MPn : termo proporcional

MIn : termo integrante

MDn : termo diferencial

Termo proporcional

$$MPn = Kc * (SPn - PVn)$$

MPn : termo proporcional

Kc : ganho

SPn : Ponto de ajuste

PVn : quantidade do processo

Termo integrante:

$$MIn = Kc * Ts * (SPn) + MIn-1$$

MIn : termo integrante

Kc : ganho

T : tempo de amostragem

TI : tempo de integração

n : Ponto de ajuste

PVn : quantidade do processo

MIn-1 : último termo integrante

Termo diferencial:

$$MDn = Kc * TD / Ts * (PVn-1n)$$

MDn : termo diferencial

Kc : ganho

TD : tempo diferencial

Ts : tempo de amostragem

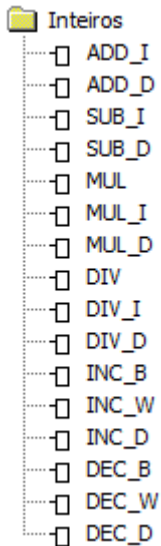
PVn-1

PVn : variável do processo

[Exemplo1](#)

[Exemplo2](#)

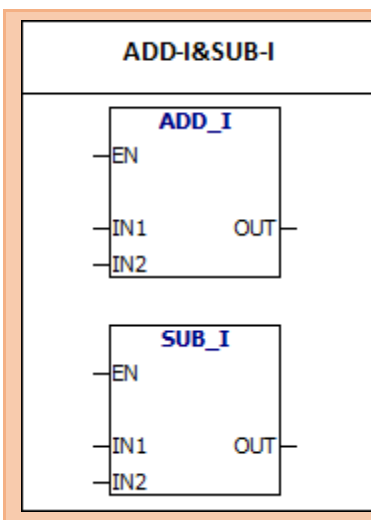
7.8. Inteiros



[Como chamar o arquivo de ajuda](#)

7.8.1. ADD-I&SUB-I

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, constant, *VD, *LD, *AC	Integer
OUT	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *LD, *AC	Integer



ADD-I: $IN1 + IN2 = OUT$ Tanto a entrada como a saída são números inteiros de 16 bits.

SUB-I: $IN1 - IN2 = OUT$ Tanto a entrada como a saída são números inteiros de 16 bits.

Em LAD e FBD: $IN1 + IN2 = OUT$

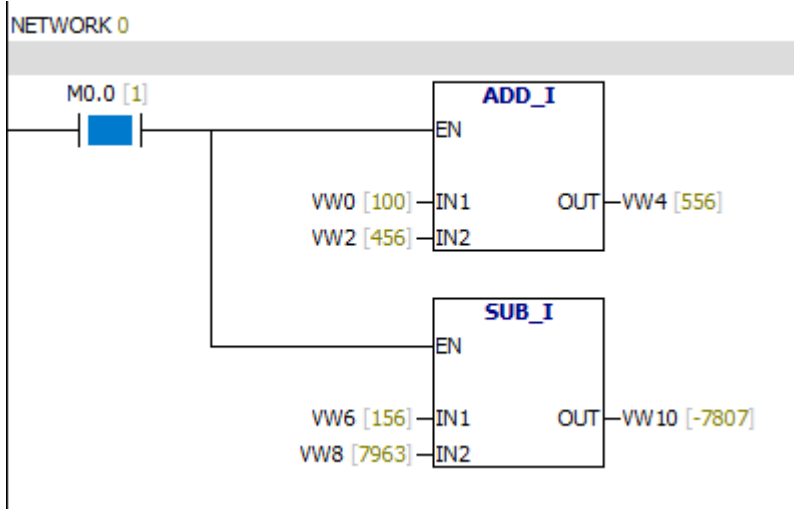
$$IN1 - IN2 = OUT$$

SM1.1 é usado para indicar erros de estouro e valores inválidos.

Condições de erro:

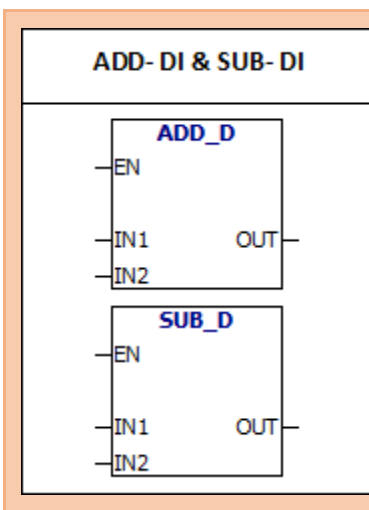
- 0006 Endereço indireto
- SM1.1 Estouro
- SM1.0 Resultado Nulo
- SM1.1 Estouro
- SM1.2 Resultado Negativo

Exemplo :



7.8.2. ADD- DI & SUB- DI

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VD, ID, QD, MD, SMD, SD, LD, AC, HC, Constant, *VD, *LD, *AC	Double integer
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Double integer



ADD- DI: $IN1 + IN2 = OUT$ Tanto a entrada como a saída são números inteiros de 32 bits.

SUB-DI: $IN1 - IN2 = OUT$ Tanto a entrada como a saída são números inteiros de 32 bits.

Em LAD e FBD: $IN1 + IN2 = OUT$
 $IN1 - IN2 = OUT$

SM1.1 é usado para indicar erros de estouro e valores inválidos.

Condições de Erro:

0006 Endereço Indireto

SM1.1 Estouro

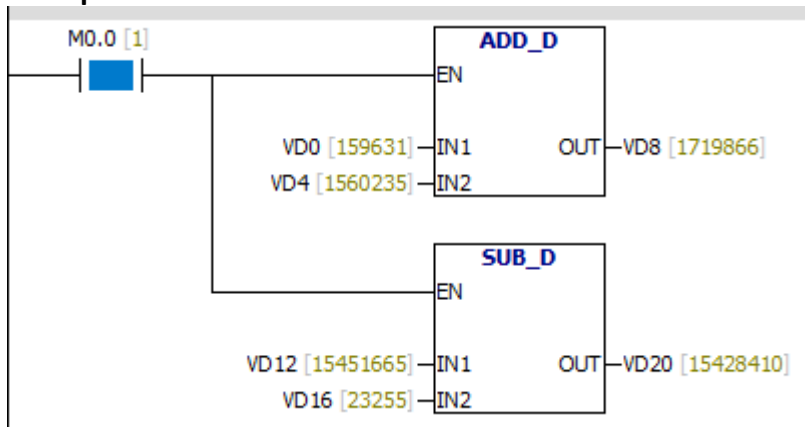
Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

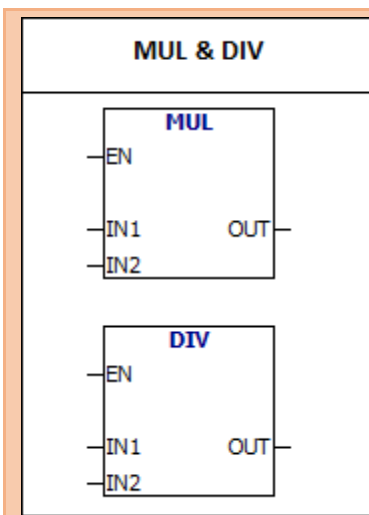
SM1.2 Resultado Negativo

Exemplo:



7.8.3. MUL & DIV

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, AIW, constant, *VD, *LD, *AC	Integer
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Double Integer



MUL: $IN1 \times IN2 = OUT$ Entrada 16 bits inteiros e saída 32 bits inteiro.

DIV: $IN1 / IN2 = OUT$ Entrada de 16 bits inteiros e o resultado de saída é de 32 bits. O resultado inclui um restante de 16 bits (alto) e um quociente de 16 bits (baixo).

Em LAD e FBD: $IN1 * IN2 = OUT$

$IN1 / IN2 = OUT$

SM1.1 é usado para indicar erros de estouro e valores inválidos.

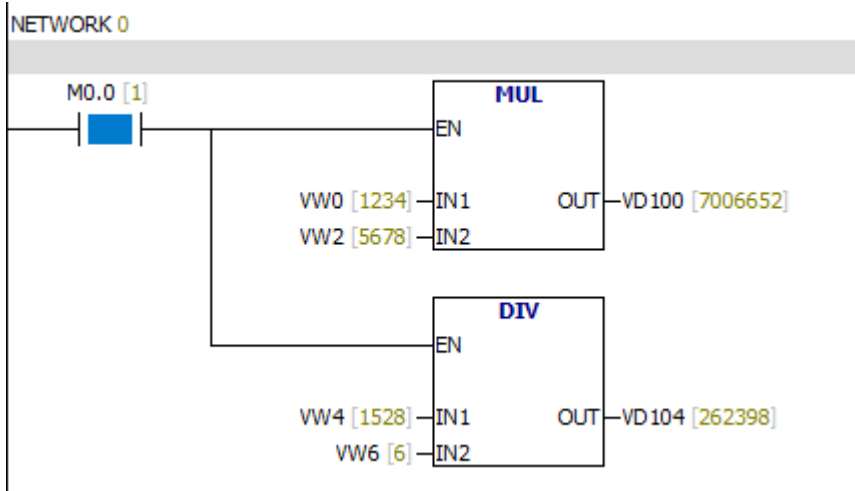
Condições de Erro:

- 0006 Endereço indireto
- SM1.1 Estouro
- SM1.3 Dividir por 0

Bit memória especial:

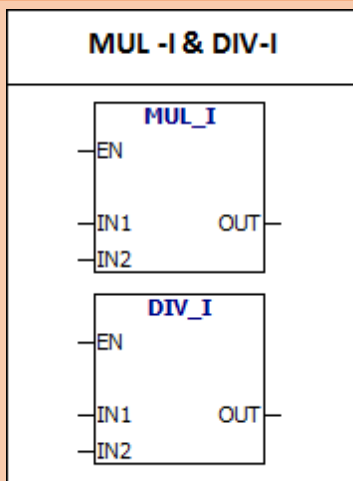
- SM1.0 Resultado Nulo
- SM1.1 Estouro

Exemplo:



7.8.4. MUL -I & DIV-I

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, AIW, constant, *VD, *LD, *AC	Integer
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	Integer



MUL -I: $IN1 * IN2 = OUT$ Tanto a entrada como a saída são inteiros de 16 bits.

DIV-I: $IN1 / IN2 = OUT$ Tanto a entrada como a saída são números inteiros de 16 bits.

Em LAD e FBD : $IN1 * IN2 = OUT$

$IN1 / IN2 = OUT$

SM1.1 é usado para indicar erros de estouro e valores inválidos.

Condições de Erro:

0006 Endereço Indireto

SM1.1 Estouro

SM1.3 Dividir por 0

Bit memória especial:

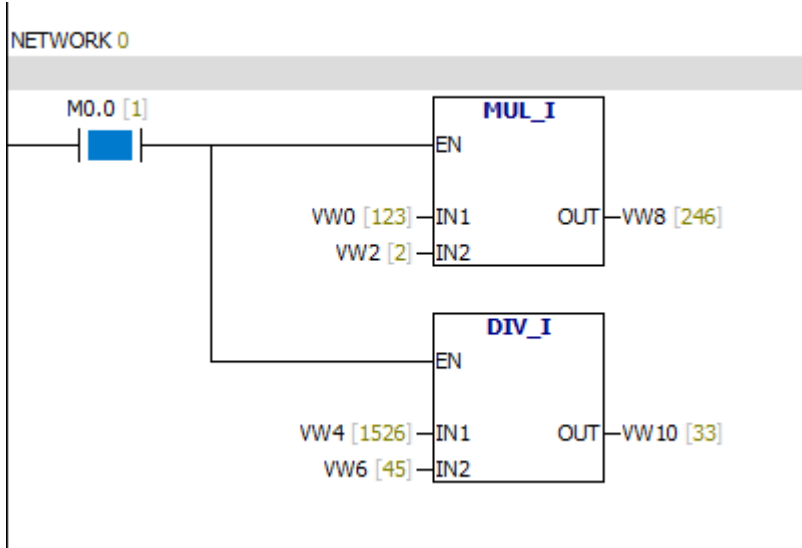
SM1.0 Resultado Nulo

SM1.1 Estouro

SM1.2 Resultado Negativo

SM1.3 Dividir por 0

Exemplo:



7.8.5. MUL -DI & DIV -DI

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VD, ID, QD, MD, SMD, SD, LD, HC, AC, constant, *VD, *LD, *AC	Double integer
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Double integer

MUL -DI & DIV -DI
<p>MUL -DI: $IN1 * IN2 = OUT$ Tanto a entrada como a saída são inteiros de 32 bits.</p> <p>DIV -DI: $IN1 / IN2 = OUT$ Tanto a entrada como a saída são 32 bits inteiros.</p> <p>A saída é um quociente. Não há resto. Em LAD e FBD: $IN1 * IN2 = OUT$</p> <p style="text-align: center;">$IN1 / IN2 = OUT$</p> <p>SM1.1 é usado para indicar erros de estouro e valores inválidos.</p>

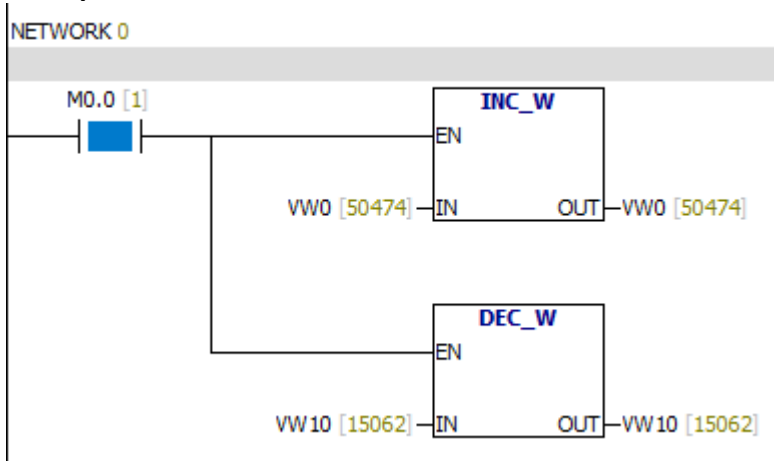
Condições de Erro:

- 0006 Endereço Indireto
- SM1.1 Estouro
- SM1.3 Dividir por 0

Bit memória especial:

- SM1.0 Resultado Nulo
- SM1.1 Estouro
- SM1.2 Resultado Negativo
- SM1.3 Dividir por 0

Exemplo:



7.8.6. INC-B & DEC-B

Entrada / Saída	Operando	Tipo de dado
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	Byte

INC-B & DEC-B	
	<p>INC-B: $IN + 1 = OUT$ Tanto a entrada como a saída são números inteiros de 8 bits.</p> <p>DEC-B: $IN - 1 = OUT$ Tanto a entrada como a saída são números inteiros de 8 bits. As duas operações de instruções não possuem símbolos.</p> <p>Em LAD e FBD : $IN + 1 = OUT$</p> <p style="text-align: center;">$IN - 1 = OUT$</p>

Condições de Erro:

0006 Endereço Indireto

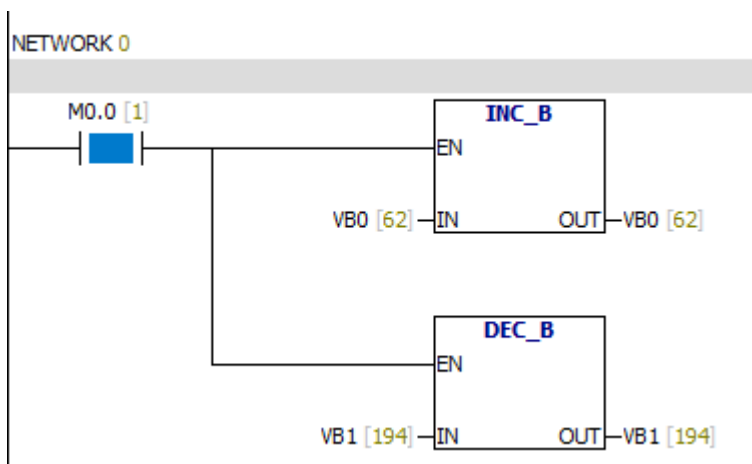
SM1.1 Estouro

Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

Exemplo:



7.8.7. INC-W & DEC-W

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, AC, AIW, LW, T, C, constant, *VD, *LD, *AC	Integer
OUT	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, *VD, *LD, *AC	Integer

Condições de Erro:

0006 Endereço Indireto

SM1.1 Estouro

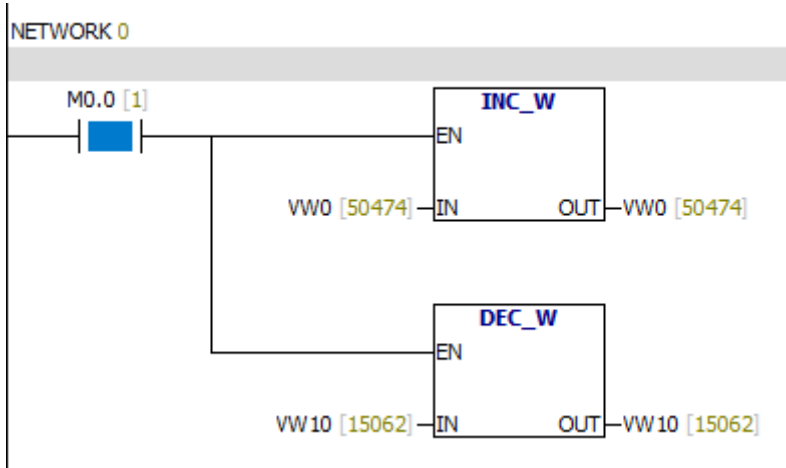
Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

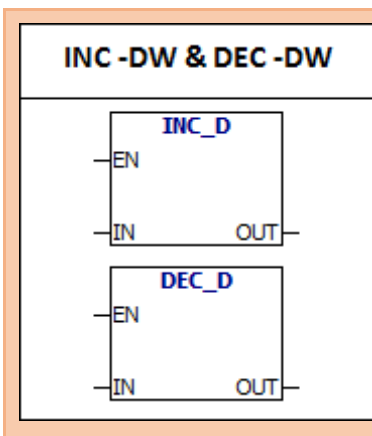
SM1.2 Resultado Negativo

Exemplo:



7.8.8. INC -DW & DEC -DW

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, AC, HC, constant, *VD, *LD, *AC	Double integer
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Double integer



INC -DW: $IN + 1 = OUT$ Tanto a entrada como a saída são inteiros duplos de 32 bits.

DEC -DW: $IN - 1 = OUT$ Tanto a entrada como a saída são 32 bits de inteiros duplos.

Em LAD e FBD: $IN + 1 = OUT$

$$IN - 1 = OUT$$

As duas instruções de operações possuem símbolos. ($16 \# 7FFFFFFF > 16 \# 80000000$).

Condições de Erro:

0006 Endereço Indireto

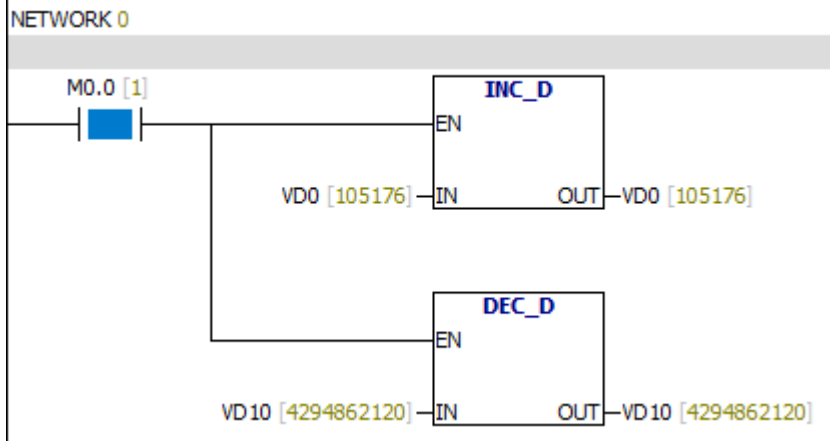
Bit memória especial:

SM1.0 Resultado Nulo

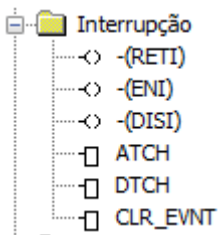
SM1.1 Estouro

SM1.2 Resultado Negativo

Exemplo:



7.9. Interrupção



[Como chamar o arquivo de ajuda](#)

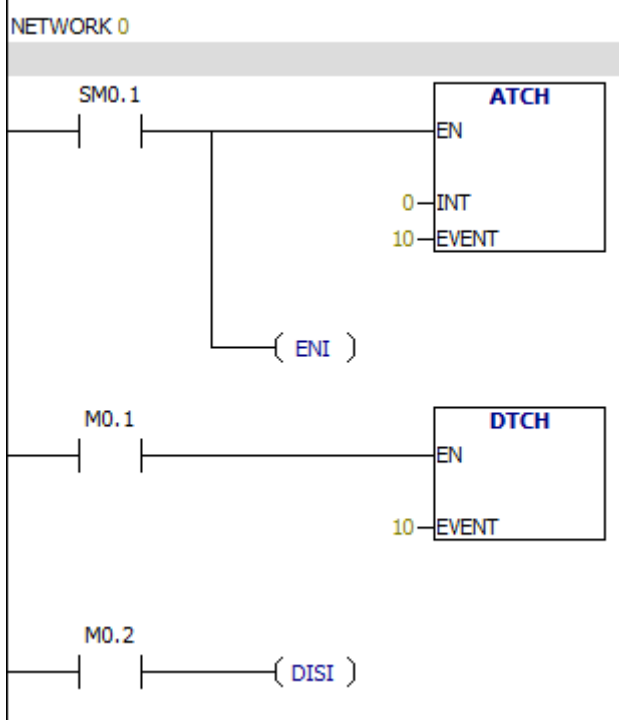
7.9.1. ENI & DISI

ENI & DISI	
	<p>Ativação de interrupção (ENI): se a instrução estiver ativada, todas as interrupções podem ser usadas.</p>
	<p>Desativação de interrupção (DISI): se a instrução estiver ativada, todas as interrupções não podem ser usadas.</p>
	<p>Quando a instrução DISI é usada, os eventos de interrupção serão enfileirados.</p>

Eventos de interrupção:

I1.2 Rising edge	PLC_EVENT_INPUTP 0	0	Prioridade muito alta
I1.4 Rising edge	PLC_EVENT_INPUTP 1	1	Prioridade alta
Timer interrupt 0	PLC_EVENT_TIMER0	10	Prioridade baixa
Timer interrupt 1	PLC_EVENT_TIMER1	11	Prioridade muito baixa

Exemplo:



7.9.2. Instrução RETI

Instrução RETI

— (RETI)

RETI: Quando há Lógica antes da instrução RETI (1), a execução do PLC retorna a partir da interrupção.

Eventos de Interrupção:

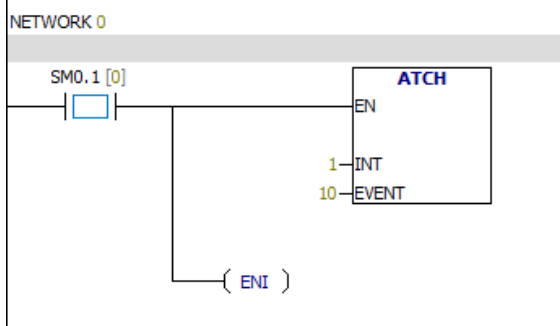
I1.2 Rising edge	PLC_EVENT_INPUTP 0	0	Prioridade muito alta
I1.4 Rising edge	PLC_EVENT_INPUTP 1	1	Prioridade alta
Timer interrupt 0	PLC_EVENT_TIMER0	10	Prioridade baixa
Timer interrupt 1	PLC_EVENT_TIMER1	11	Prioridade muito baixa

Guia de interrupção

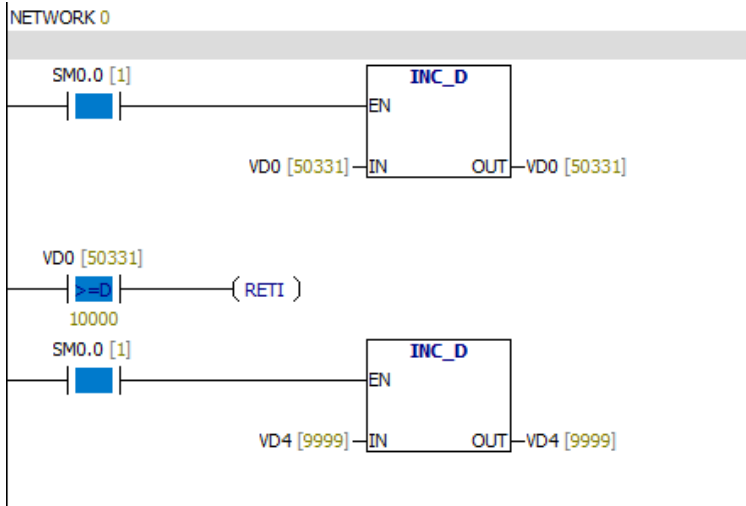
A rotina de interrupção oferece uma resposta rápida a um evento interno ou externo particular. A rotina de interrupção deve ser concisa e eficiente, de modo que pode acelerar a velocidade de execução.

Limite: as instruções DISI, ENI, HDEF, LSCR e END não podem ser usadas na rotina de interrupção.

Exemplo:

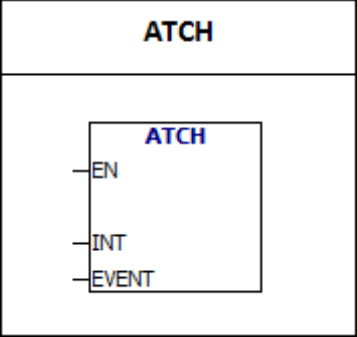


Programa de Interrupção:



7.9.3. ATCH

Entrada / Saída	Operando	Tipo de dado
INT	Constant 0-127	Byte
EVNT	Constant 0-33	Byte

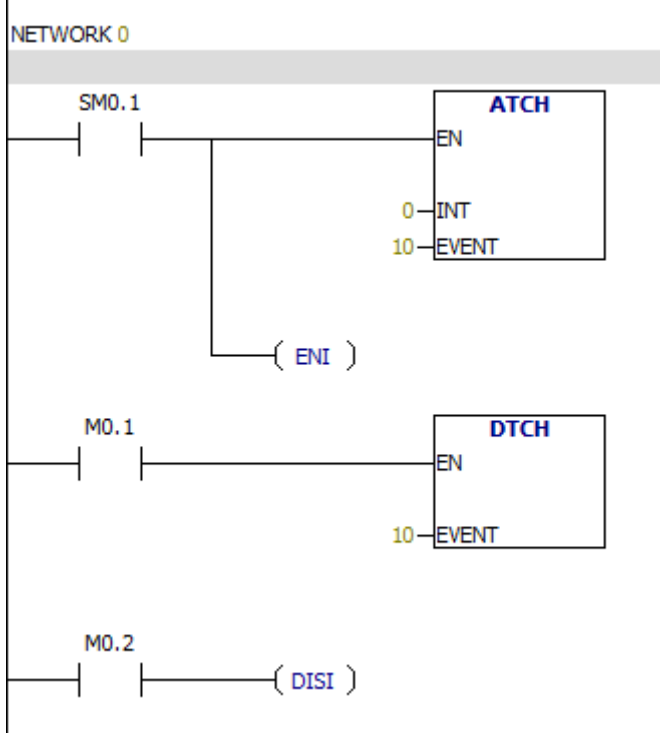
	<p>ATCH: O evento de interrupção (EVNT) está conectado ao número de rotina de interrupção (INT) pela instrução "ATCH" e, em seguida, ativa o evento de interrupção.</p>
---	--

Você pode anexar mais de um evento de interrupção a uma rotina de interrupção. No entanto, um evento de interrupção não pode ser anexado às rotinas de interrupção múltiplas. Quando você anexa um evento de interrupção a uma rotina de interrupção, a interrupção é ativada automaticamente. Quando a instrução DISI é usada, os eventos de interrupção serão enfileirados. Se você quiser desativar um único evento de interrupção, você pode usar as instruções "DTCH".

Eventos de Interrupção:

I1.2 Rising edge	PLC_EVENT_INPUTP 0	0	Prioridade muito alta
I1.4 Rising edge	PLC_EVENT_INPUTP 1	1	Prioridade alta
Timer interrupt 0	PLC_EVENT_TIMER0	10	Prioridade baixa
Timer interrupt 1	PLC_EVENT_TIMER1	11	Prioridade muito baixa

Exemplo:



As instruções "WATCH" só precisam ser conectadas uma vez.

7.9.4. DTCH

Entrada / Saída	Operando	Tipo de dado
EVNT	Constant (0-33)	Byte

DTCH

A instrução de separação de interrupção (DTCH) cancela a associação entre o evento de interrupção (EVNT) e a rotina de interrupção e desativa o evento de interrupção.

O evento de interrupção (EVNT) está conectado ao número de rotina de interrupção (INT) pela instrução "ATCH" e em seguida, ativa o evento de interrupção.

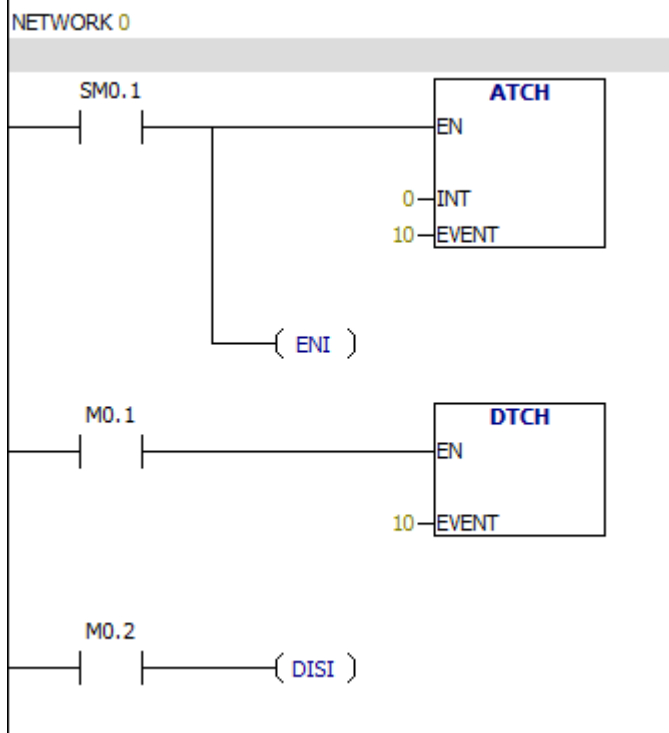
Você pode anexar mais de um evento de interrupção a uma rotina de interrupção. No entanto, um evento de interrupção não pode ser anexado às rotinas de interrupção múltiplas. Quando você anexa um evento de interrupção a uma rotina de interrupção, a interrupção é ativada automaticamente. Quando a instrução DISI é Usado, os eventos de interrupção serão enfileirados.

Se você quiser desativar um único evento de interrupção, você pode usar as instruções "DTCH".

Eventos de Interrupção:

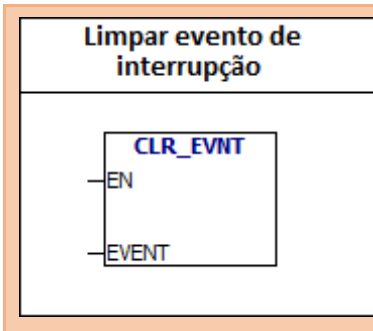
I1.2 Rising edge	PLC_EVENT_INPUTP 0	0	Prioridade muito alta
I1.4 Rising edge	PLC_EVENT_INPUTP 1	1	Prioridade alta
Timer interrupt 0	PLC_EVENT_TIMER0	10	Prioridade baixa
Timer interrupt 1	PLC_EVENT_TIMER1	11	Prioridade muito baixa

Exemplo:



7.9.5. Limpar evento de interrupção

Entrada / Saída	Operando	Tipo de dado
EVNT	Constante	Byte

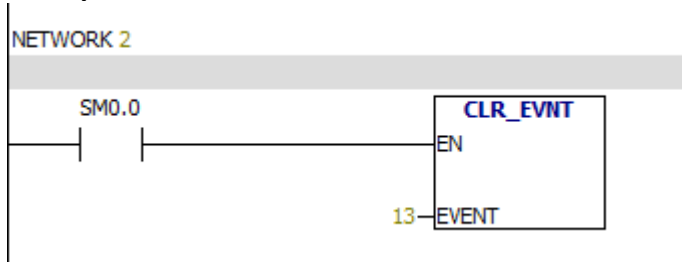


CLR - EVNT A instrução removerá todos os tipos EVNT de interrupção de evento na fila de interrupção. Esta instrução é usada para remover interrupções desnecessárias.

Interrupt Events:

I1.2 Rising edge	PLC_EVENT_INPUTP 0	0	Prioridade muito alta
I1.4 Rising edge	PLC_EVENT_INPUTP 1	1	Prioridade alta
Timer interrupt 0	PLC_EVENT_TIMER0	10	Prioridade baixa
Timer interrupt 1	PLC_EVENT_TIMER1	11	Prioridade muito baixa

Exemplo:



7.10. Lógica binária

- Logica Binária
 - INV_B
 - INV_W
 - INV_DW
 - WAND_B
 - WAND_W
 - WAND_D
 - WOR_B
 - WOR_W
 - WOR_D
 - WXOR_B
 - WXOR_W
 - WXOR_DW

7.10.1. INV-B

Entrada / Saída	Operando	Tipo de dado
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *AC, *LD	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	Byte

INV -B

INV -B: A instrução executa a operação do complemento para o byte de entrada e coloca o resultado em OUT.

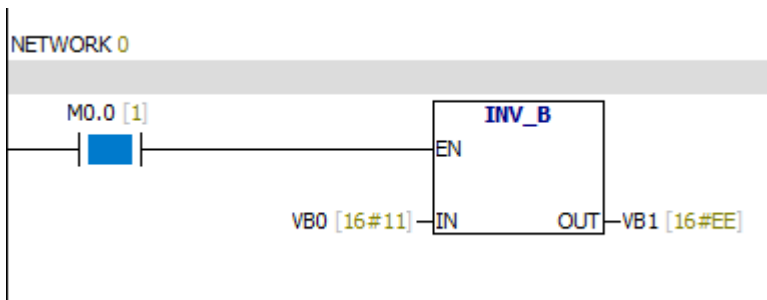
Condição de Erro:

0006 Endereço Indireto

Bit memória especial:

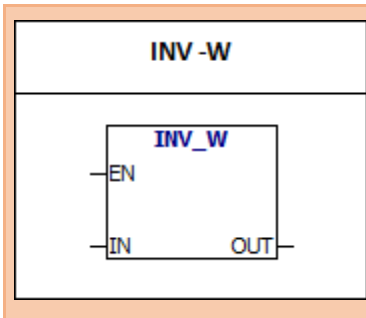
SM1.0 Resultado Nulo

Exemplo:



7.10.2. INV-W

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, T, C, AIW, LW, AC, constant, *VD, *AC, *LD	Word
OUT	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD	Word



INV -W: A instrução executa a operação do complemento para a palavra de entrada e coloca o resultado em OUT.

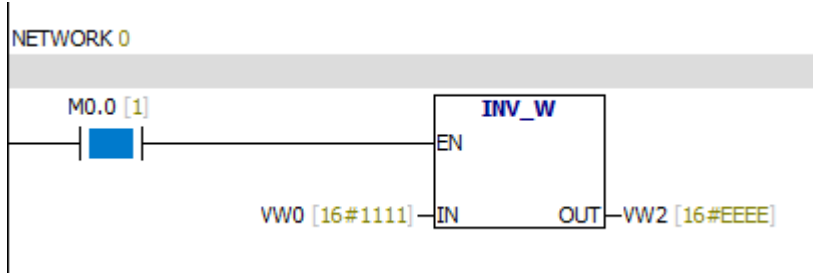
Condição de Erro:

0006 Endereço Indireto

Bit memória especial:

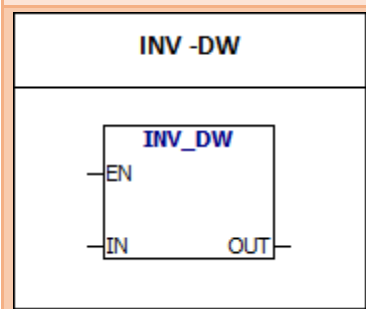
SM1.0 Resultado Nulo

Exemplo:



7.10.3. INV-DW

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, HC, AC, constant, *VD, *AC, *LD	Double word
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	Double word



INV -DW: A instrução executa a operação do complemento para a word de entrada e coloca o resultado em OUT.

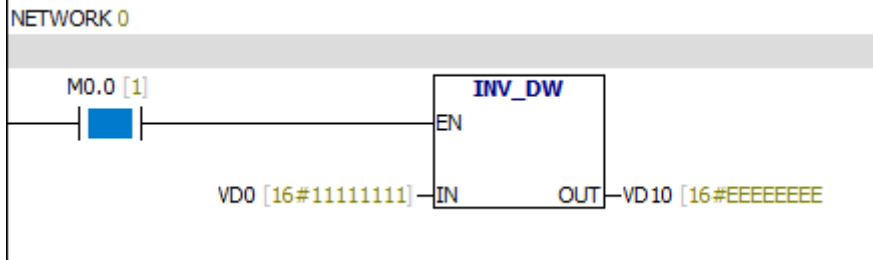
Condição de Erro:

0006 Endereço Indireto

Bit memória especial:

SM1.0 Resultado Nulo

Exemplo:



7.10.4. WAND-B, WOR -B, WXOR -B

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *AC, *LD	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	Byte

WAND-B、WOR-B、 WXOR-B	
	WAND -B: Executa a instrução "AND" em IN1 e IN2. Então coloca o resultado em OUT.
	WOR -B: Executa a instrução "OR" em IN1 e IN2. Então coloca o resultado em OUT.
	WXOR -B: Executa a instrução o "XOR" em IN1 e IN2. Então coloca o resultado em OUT.

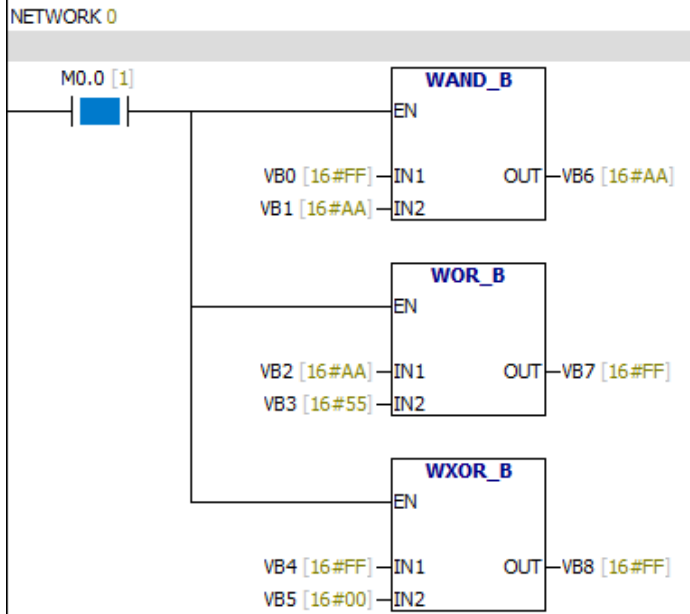
Condição de Erro:

0006 Endereço Indireto

Bit memória especial:

SM1.0 Resultado Nulo

Exemplo:



7.10.5. WAND-W, WOR -W, WXOR -W

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, constant, *VD, *AC, *LD	Word
OUT	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD	Word
<div style="border: 1px solid black; padding: 5px;"> <p>WAND-W、WOR-W、WXOR-W</p> <div style="display: flex; flex-direction: column; gap: 10px;"> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">WAND_W</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">WOR_W</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">WXOR_W</p> </div> </div> </div>		
		WAND -W: Executa a instrução "AND" em IN1 e IN2. Então coloca o resultado em OUT.
		WOR -W: "OR" em IN1 e IN2. Então coloca o resultado em OUT.
		WXOR -W: Executa a instrução "XOR" em IN1 e IN2. Então coloca o resultado em OUT.

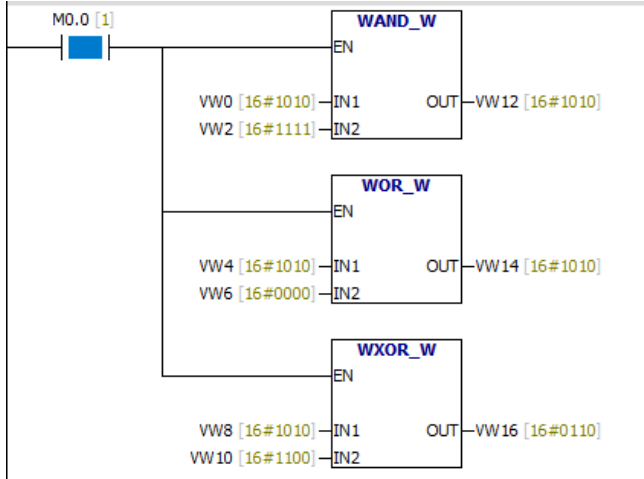
Condição de Erro:

0006 Endereço Indireto

Bit memória especial:

SM1.0 Resultado Nulo

Exemplo:



7.10.6. WAND- DW, WOR -DW, WXOR -DW

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VD, ID, QD, MD, SMD, AC, LD, HC, constant, *VD, *AC, SD, *LD	Double word
OUT	VD, ID, QD, MD, SMD, LD, AC, *VD, *AC, SD, *LD	Double word

WAND- DW \ WOR -DW \ WXOR -DW	
	<p>WAND -DW: Executa a instrução "AND" em IN1 e IN2. Então coloca o resultado em OUT.</p>
	<p>WOR -DW: Executa a instrução "OR" em IN1 e IN2. Então coloca o resultado em OUT.</p>
	<p>WXOR -DW: Executa a instrução "XOR" em IN1 e IN2. Então coloca o resultado em OUT.</p>

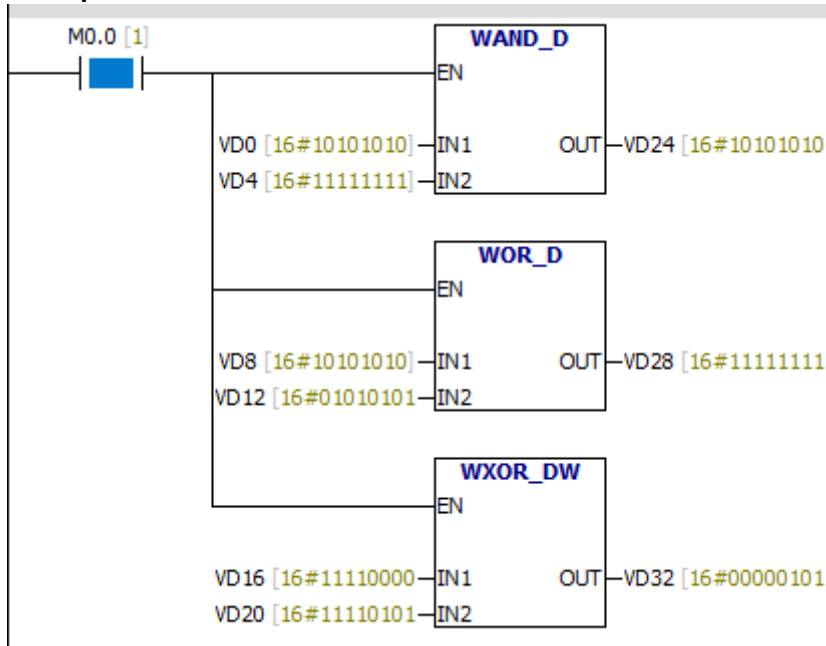
Condição de Erro:

0006 Endereço Indireto

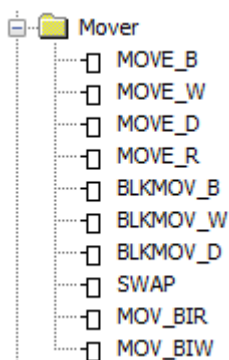
Bit memória especial:

SM1.0 Resultado Nulo

Exemplo:

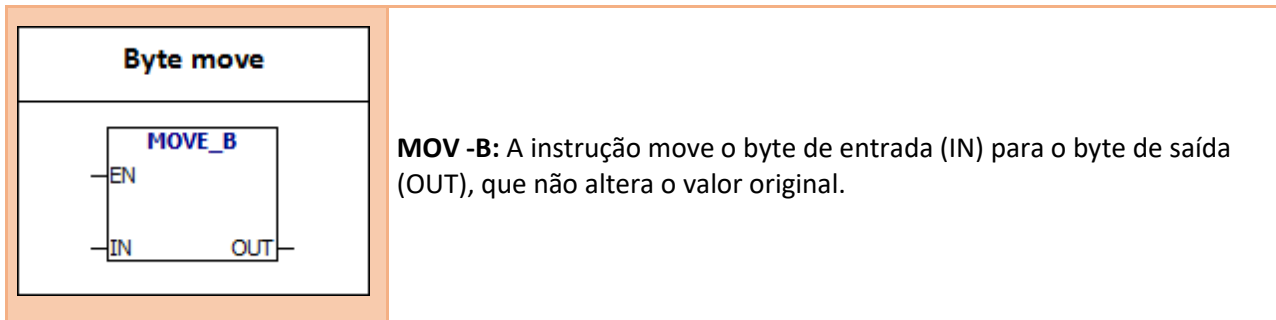


7.11. Mover



7.11.1. Move Byte

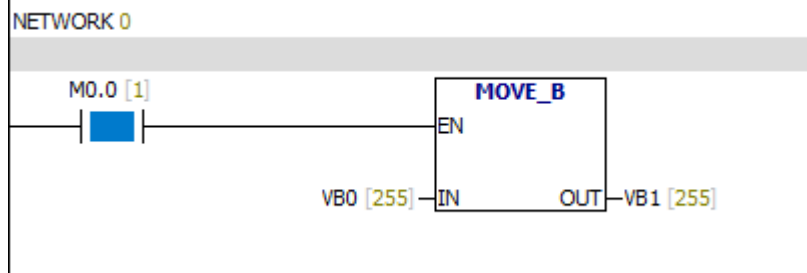
Entrada / Saída	Operando	Tipo de dado
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	Byte



Condição de Erro:

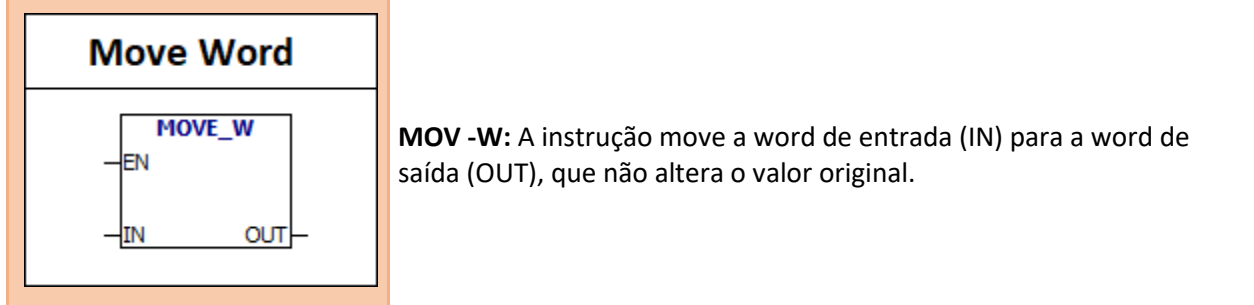
0006 Endereço Indireto

Exemplo:



7.11.2. Move Word

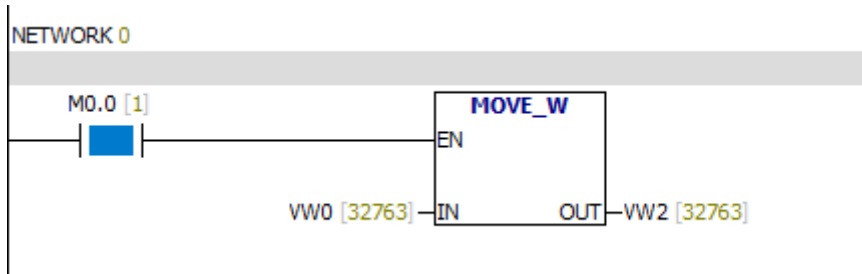
Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, constant, AC, *VD, *AC, *LD	word, integer
OUT	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	word, integer



Condição de Erro:

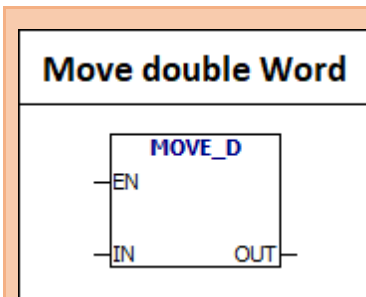
0006 Endereço Indireto

Exemplo:



7.11.3. Move Double word

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, HC, &VB, &IB, &QB, &MB, &SB, &T, &C, &SMB, &AIW, &AQW AC, constant, *VD, *LD, *AC	Double word, double integer
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Double word, double integer

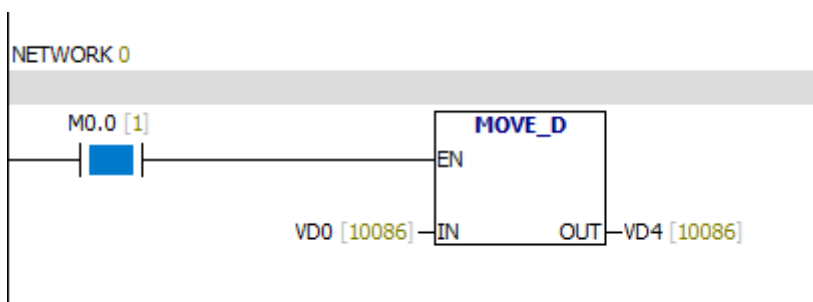


MOV -DW: A instrução move a double word de entrada (IN) para a double word de saída (OUT), que não altera o valor original. Você pode usar a instrução "MOVE-D" para criar um ponteiro.

Condição de Erro:

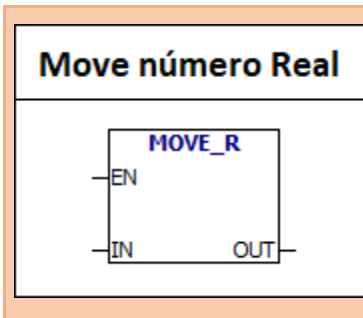
0006 Endereço Indireto

Exemplo:



7.11.4. Move Número real

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SD, SMD, LD, AC, constant, *VD, *LD, *AC	Número Real
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Número Real

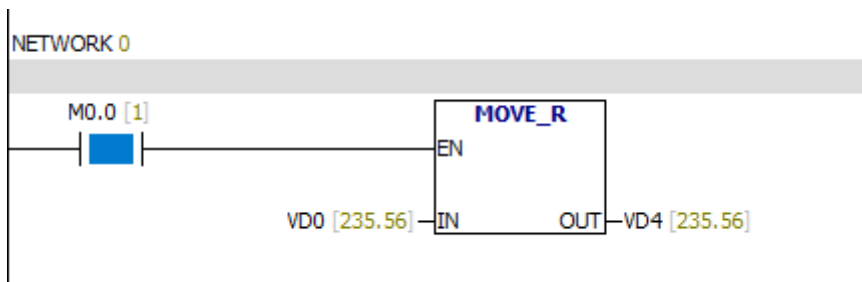


MOV -R: A instrução move à entrada Número Real (IN) para a saída Número Real (OUT), que não altera o valor original.

Condição de Erro:

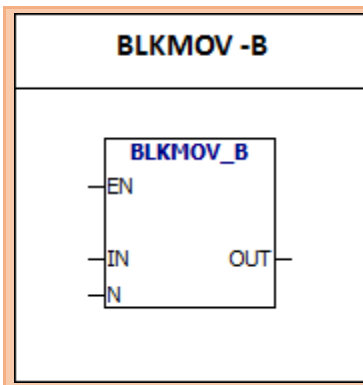
0006 Endereço Indireto

Exemplo:



7.11.5. BLKMOV -B

Entrada / Saída	Operando	Tipo de dado
IN	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD	Byte
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *AC, *LD	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, *VD, *AC, *LD	Byte



BLKMOV-B: Estes sucessivos bytes "N" que começam com "IN" são movidos para OUT.

O intervalo de N é de 1 a 255.

Condições de Erro:

0006 Endereço Indireto

Exemplo:

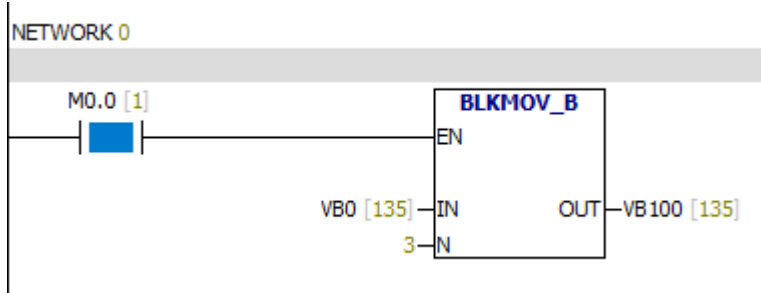
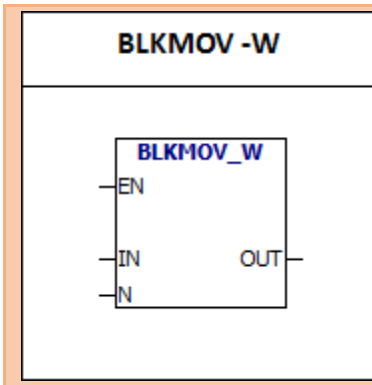


Tabela de monitoramento:

Address	Data Type	Value
M0.0	BOOL	1
VB0	USINT	135
VB1	USINT	153
VB2	USINT	255
VB100	USINT	135
VB101	USINT	153
VB102	USINT	255

7.11.6. BLKMOV-W

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, *VD, *LD, *AC	Word
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *LD, *AC	Word



BLKMOV -W: Estas sucessivas word "N" que começam com "IN" são movidas para OUT.

O intervalo de N é de 1 a 255.

Condições de Erro:

0006 Endereço Indireto

0091 O número operacional está fora do alcance

Exemplo:

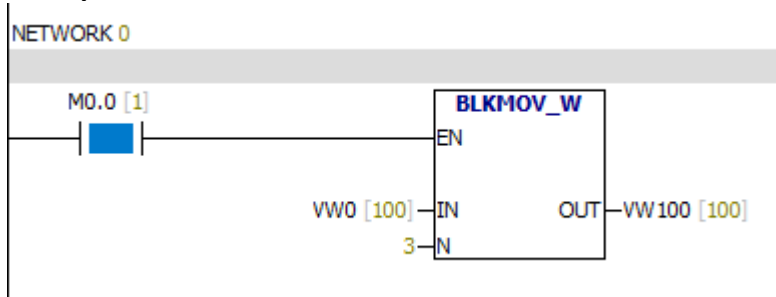
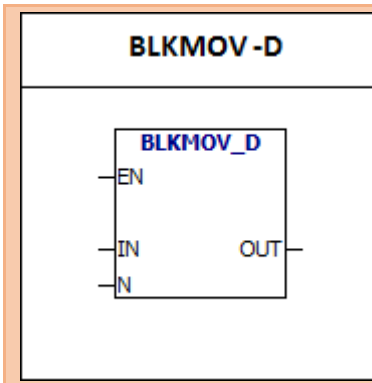


Tabela de monitoramento:

	Address	Data Type	Value
	M0.0	BOOL	1
	VW0	INT	100
	VW2	INT	101
	VW4	INT	102
	VW100	INT	100
	VW102	INT	101
	VW104	INT	102

7.11.7. BLKMOV -D

Entrada / Saída	Operando	Tipo de dado
IN, OUT	VD, ID, QD, MD, SD, SMD, LD, *VD, *AC, *LD	Double word
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *AC, *LD	Byte



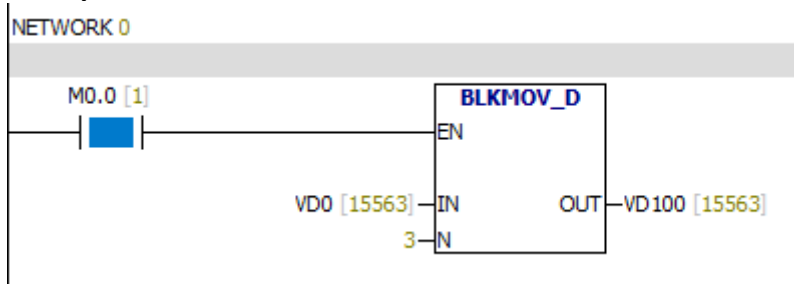
BLKMOV - D: Estas sucessivas double words "N" que começam com "IN" são movidas para OUT.
O intervalo de N é de 1 a 255.

Condições de Erro:

0006 Endereço Indireto

0091 Operating number is out of range

Exemplo:



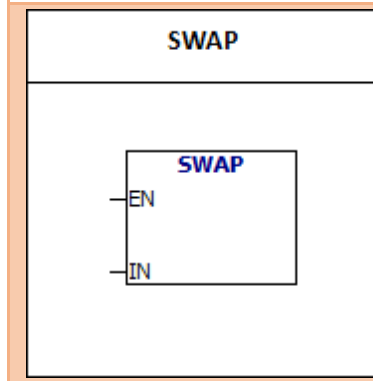
VD0=15563 VD100=15563

VD4=15573 VD104=15573

VD8=15583 VD108=15583

7.11.8. SWAP

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, T, C, LW, AC, *VD, *AC, *LD	Word



SWAP: a instrução troca o primeiro byte e o segundo byte da word de entrada.

Condições de Erro:

0006 Endereço Indireto

Exemplo: Quando M0.0=0;

NETWORK 0

MAIN (INT0) INT_1 (INT1) SBR_0 (SBR0)

Chart

Address	Data Type	Value	Forced	Address
M0.0	BOOL	0		
VV0	WORD	16#ABCD		

Quando M0.0=1;

NETWORK 0

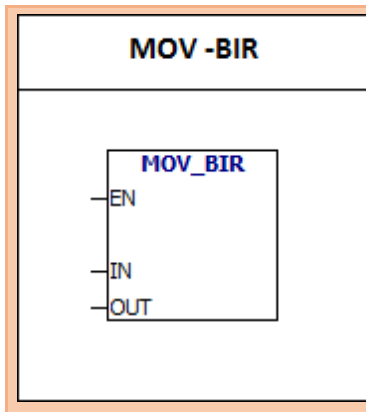
MAIN (INT0) INT_1 (INT1) SBR_0 (SBR0)

Chart

Address	Data Type	Value	Forced	Address
M0.0	BOOL	1		
VV0	WORD	16#CDAB		

7.11.9. MOV-BIR

Entrada / Saída	Operando	Tipo de dado
IN	IB, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	Byte



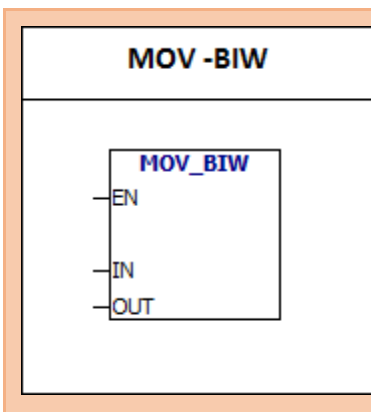
MOV -BIR: a instrução lê o valor real de entrada (byte), em seguida, grava o valor em OUT. O registro da imagem do processo não é atualizado.

Condições de Erro:

0006 Endereço Indireto

7.11.10. MOV-BIW

Entrada / Saída	Operando	Tipo de dado
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *AC, *LD	Byte
OUT	QB, *VD, *LD, *AC	Byte

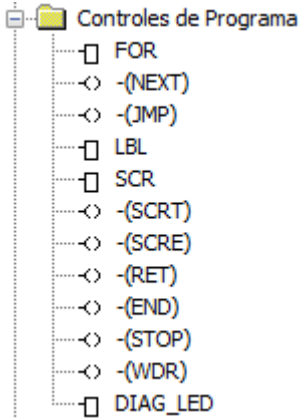


MOV -BIW: A instrução grava o valor de entrada (IN) na entrada real (OUT) e atualiza o registro de imagem de processo correspondente.

Condições de Erro:

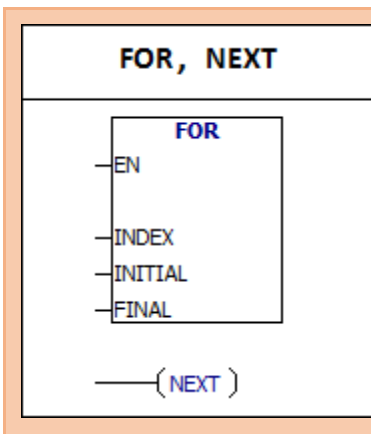
0006 Endereço Indireto

7.12. Controles de Programa



7.12.1. FOR, NEXT

Entrada / Saída	Operando	Tipo de dado
INDX	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	Integer
INIT	VW, IW, QW, MW, SW, SMW, T, C, AC, LW, AIW, constant, *VD, *LD, *AC	Integer
FINAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, AIW, constant, *VD, *LD, *AC	Integer

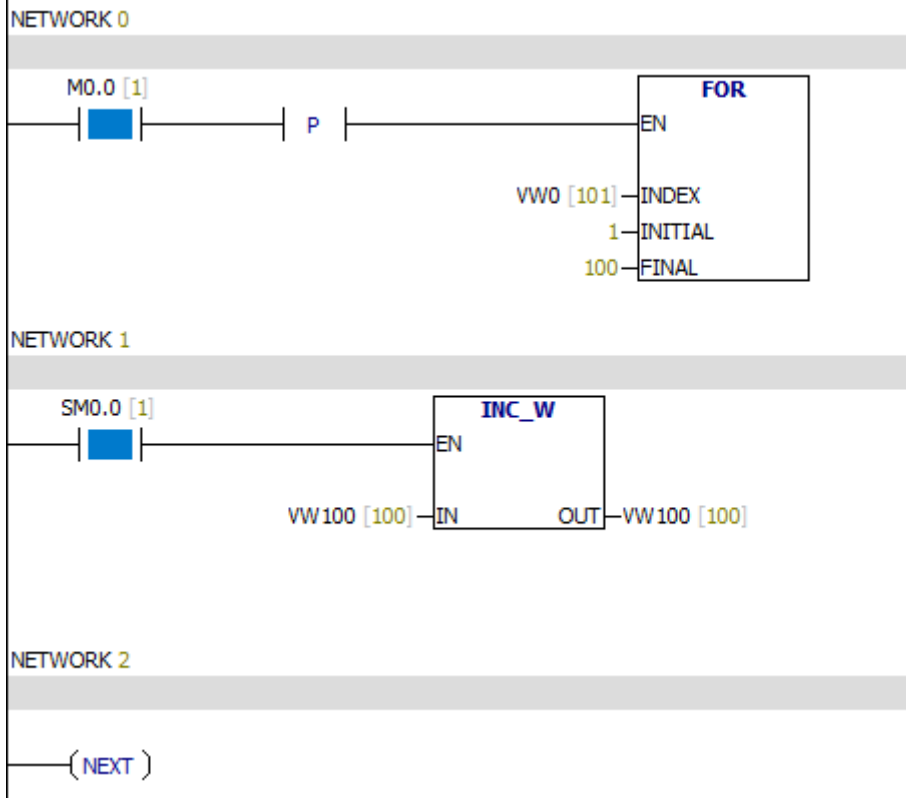


A instrução FOR executa instruções entre FOR e NEXT. Você precisa especificar a contagem de ciclo atual (INDX), o valor de início (INIT) e o valor final (FINAL). A instrução next (NEXT) marca o fim do loop FOR e o topo, o valor da pilha está configurado para 1. Use FOR / NEXT para definir o número de loops. Cada instrução FOR requer uma instrução NEXT. Os laços FOR / NEXT podem ser alinhados com 8 Loops FOR / NEXT. Após a execução de cada Instrução do FOR e NEXT, o valor INDX é aumentado e o resultado é comparado com o valor final. Se o INDX for maior do que o valor final, o ciclo termina.

Condição de Erro:

0006 Endereço Indireto

Exemplo:

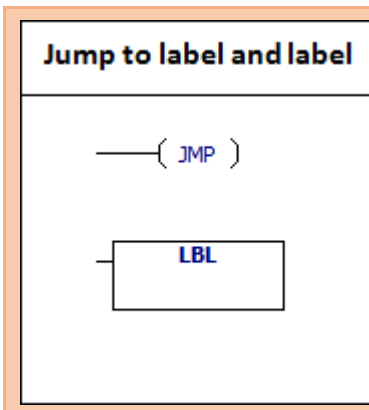


Nota :

Os tempos de ciclo são configurados para 100 vezes. No final do ciclo, o valor de VW100 é 100.

7.12.2. Pular

Entrada / Saída	Tipo de dado
n: constant (0~255)	Word



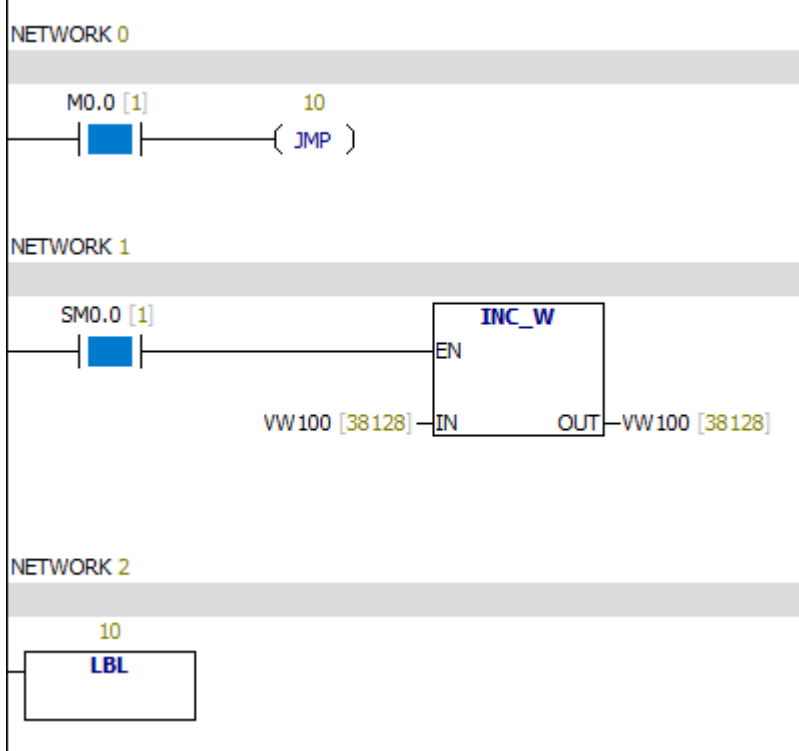
A instrução JMP executa a operação de ramificação para o programa na tag especificada (n). Quando o salto é aceito, o valor superior da pilha é 1.

A instrução LBL assina a localização de n.

Você pode usar as instruções de "JUMP" no programa principal, na sub-rotina ou na rotina de interrupção.

Você não pode pular do programa principal para uma sub-rotina ou uma rotina de interrupção. Você pode usar a instrução "JUMP" no segmento SCR, mas a instrução correspondente "TAG" deve estar localizada dentro do mesmo segmento SCR.

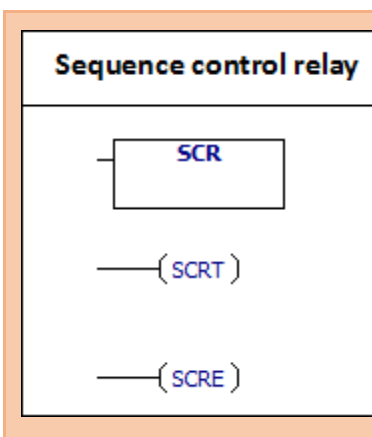
Exemplo:



Quando o bit M0.0 é 1, o valor de VW100 não é mais aumentado.

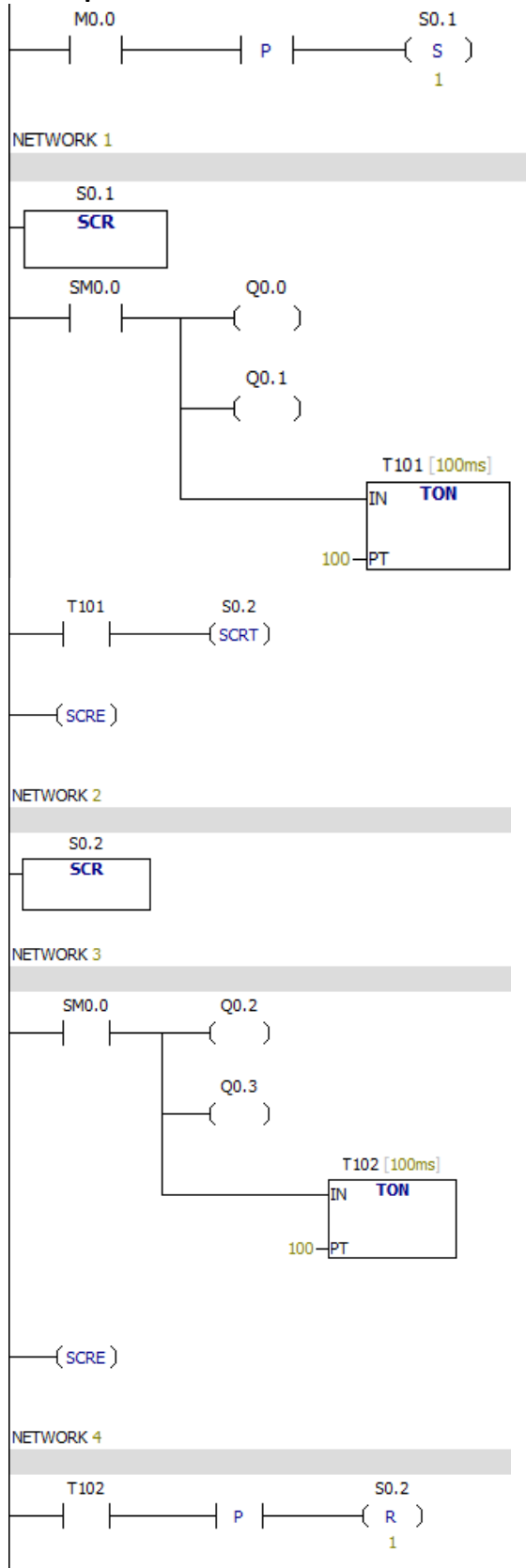
7.12.3. Relé de controle de sequência

Entrada / Saída	Operando	Tipo de dado
n	S	Booleana

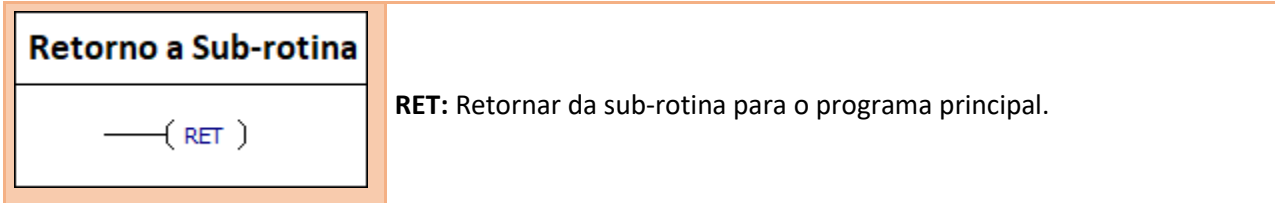


A instrução SCR é boa ao lidar com operações repetitivas.
 SCR: carregue a seção SCR, você pode usar as instruções SET.
 SCRT: Ir para outro segmento SCR e fechar o segmento SCR atual.
 SCRE: A instrução assina o fim do segmento SCR.

Exemplo:

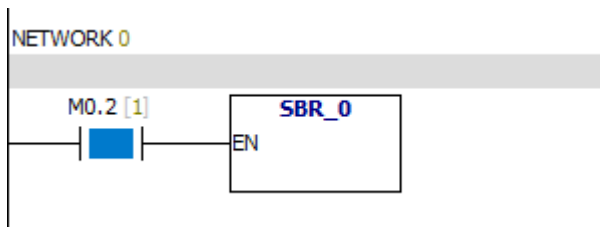


7.12.4. Retorno da sub-rotina

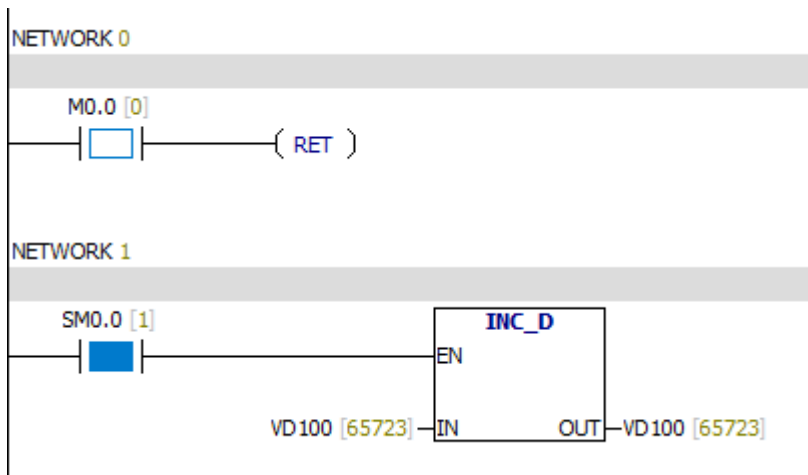


Exemplo:

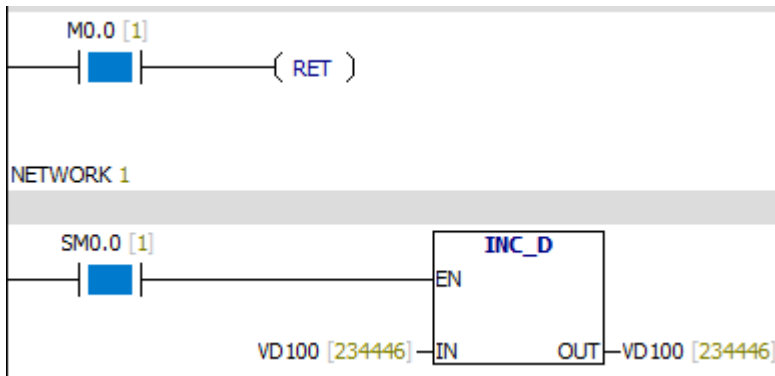
Programa principal:



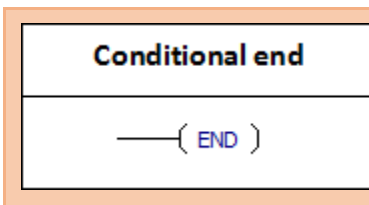
Sub-rotina:



Quando o bit M0.0 é 1, retornar da sub-rotina, o programa a seguir não será mais escaneado.



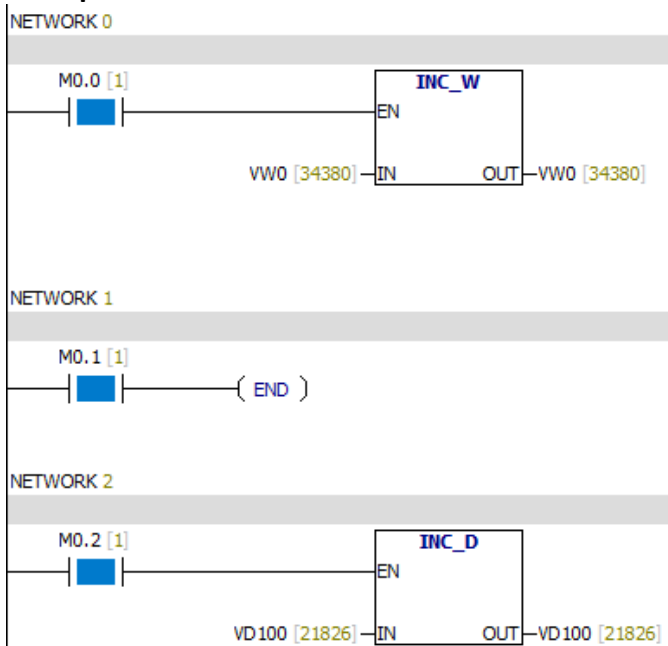
7.12.5. Fim condicional



A instrução END encerra o programa do usuário.

Notas: Você pode usar as instruções "END" no programa principal, mas não pode ser usado na rotina de sub-rotina ou interrupção.

Exemplo:



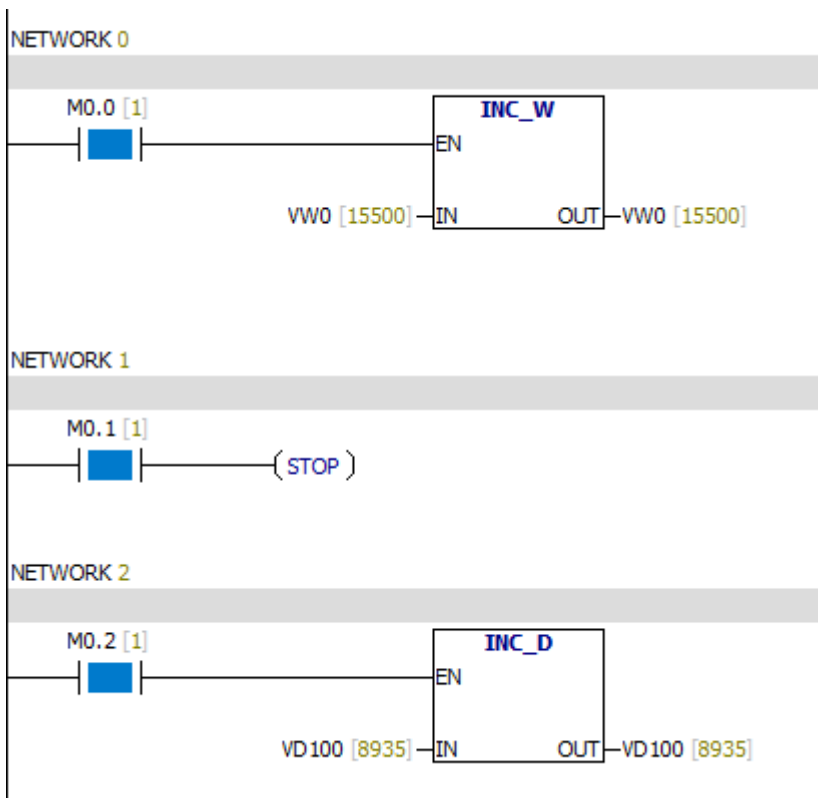
Quando o bit M0.1 é 1, o programa não será digitalizado.

7.12.6. STOP

STOP

Intrusão de STOP

Exemplo:



Quando o bit M0.1 é 1, o PLC converte-se para o modo STOP, todos os programas param de funcionar.

7.12.7. Watchdog Reset

Watchdog Reset

—(WDR)

WDR reinicia o tempo de watchdog. Quando o ciclo de varredura é maior do que o tempo de watchdog, o WDR faz o watchdog não emitir um aviso.

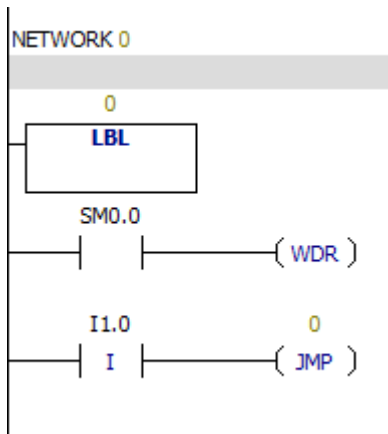
Usar instruções "WDR" deve ser usada cuidadosamente.

Os seguintes programas podem ser realizados após o ciclo de verificação ser concluído:

1. Comunicação
2. Atualiza a E/S (exceto E/S imediata)
3. Atualização forçada
4. Atualização dos bits SM
5. Programa de diagnóstico de tempo de execução
6. STOP (stop) instrução para rotina de interrupção

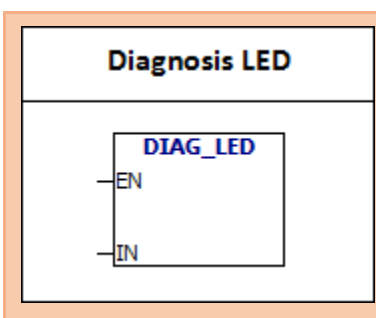
Atenção: Se você espera que o tempo de varredura seja superior a 500 ms, você deve usar a instrução WDR para ativar o temporizador watchdog.

Exemplo:



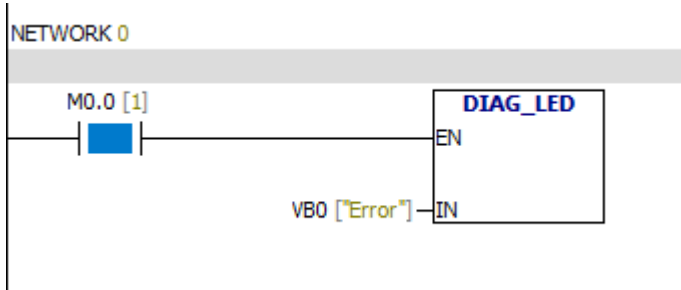
7.12.8. Diagnosis LED

Entrada / Saída	Operando	Tipo de dado
IN	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	String



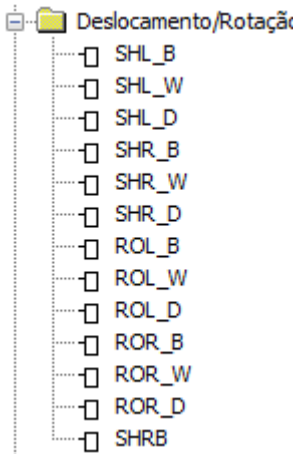
Se o valor de "EN" for 1, o LCD exibirá a string de "IN".

Exemplo:



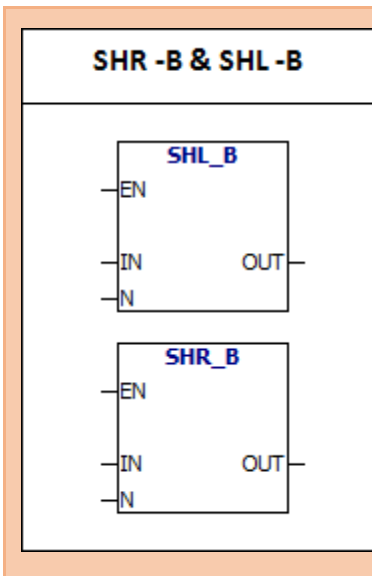
Quando o valor de M0.0 é igual a 1, o LCD exibirá "Erro".

7.13. Deslocamento / Rotação



7.13.1. SHR -B & SHL -B

Entrada / Saída	Operando	Tipo de dado
IN (LAD, FBD)	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	Byte



SHR -B: Entra com o byte "IN" e move N bits para a direita. Em seguida, coloca os resultados em OUT.

SHL -B: Entra com o byte "IN" e move N bits para a esquerda. Em seguida, coloca os resultados em OUT.

Os bits movidos são preenchidos com zero. A instrução pode mover até 8 bits.

As operações SHR-B & SHL -B não são assinadas.

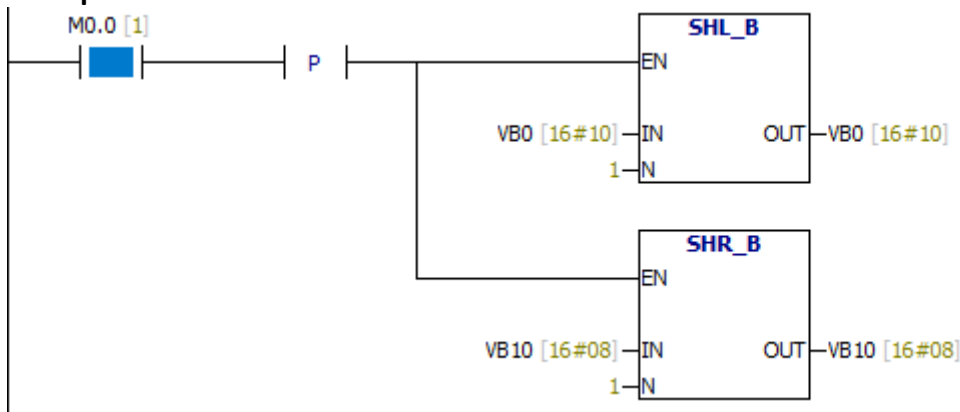
Condições de Erro:

0006 Endereço Indireto

Bit memória especial:

SM1.0 Resultado Nulo

Exemplo:



Quando o valor de M0.0 é 1, VB0 move um pouco para a esquerda e VB10 move um pouco para a direita.

7.13.2. SHR -W & SHL -W

Entrada / Saída	Operando	Tipo de dado
IN (LAD, FBD)	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constant, *VD, *LD, *AC	Word
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	Word

SHR -W & SHL -W

SHL_W

SHR_W

SHR -W: word de entrada "IN" e move N bits para a direita. Em seguida, coloque os resultados em OUT.

SHL -W: word de entrada "IN" e move N bits para a esquerda. Em seguida, coloca os resultados em OUT.

Os bits movidos são preenchidos com zero. A instrução pode mover até 16 bits. As operações SHR-W & SHL-W são assinadas. O bit de símbolo pode ser movido.

Condições de Erro:

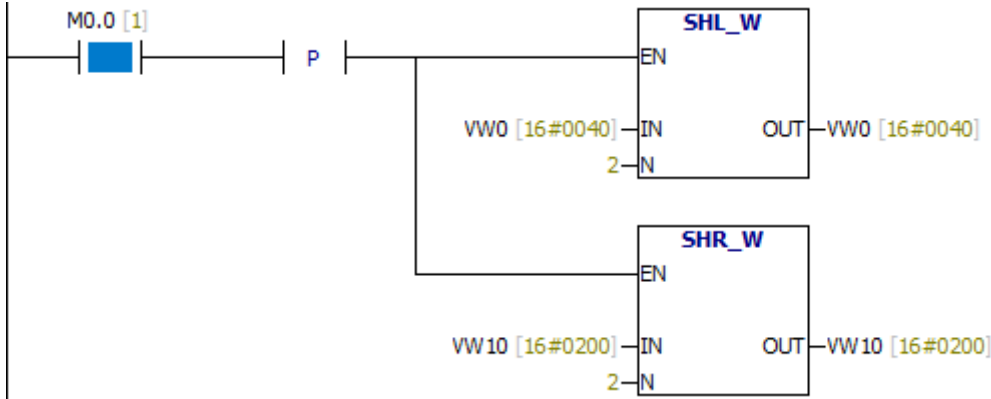
0006 Endereço Indireto

Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

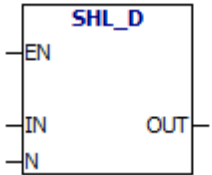
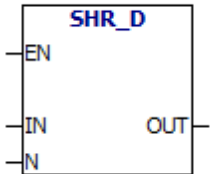
Exemplo:



Quando o valor de M0.0 é 1, VV0 move dois bits para a esquerda e VV10 move dois bits para a direita.

7.13.3. SHR -DW & SHL -DW

Entrada / Saída	Operando	Tipo de dado
IN (LAD, FBD)	VD, ID, QD, MD, SD, SMD, LD, AC, HC, constant, *VD, *LD, *AC	Double word
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Double word
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Double word

SHR -DW & SHL -DW	
	<p>SHR -DW: double word de entrada "IN" move N bits para a direita. Em seguida, coloca os resultados em OUT.</p>
	<p>SHL -DW: double word de entrada "IN" move N bits para a esquerda. Em seguida, coloca os resultados em OUT.</p>
<p>Os bits movidos são preenchidos com zero. A instrução pode mover até 32 bits. As operações SHR -DW & SHL -DW são assinadas. O bit de símbolo pode ser movido.</p>	

Condições de Erro:

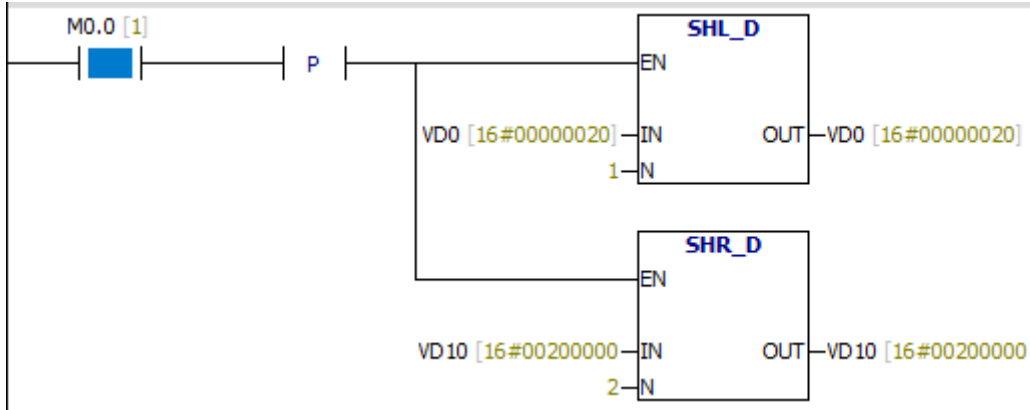
0006 Endereço Indireto

Bit memória especial:

SM1.0 Resultado Nulo

SM1.1 Estouro

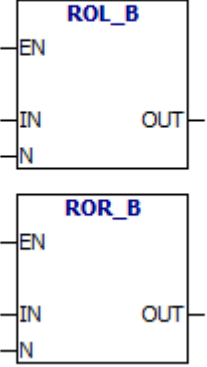
Exemplo:



Quando o valor de M0.0 é 1, VD0 move um pouco para a esquerda e VD10 move dois bits para a direita.

7.13.4. ROR -B & ROL -B

Entrada / Saída	Operando	Tipo de dado
IN (LAD, FBD)	VB, IB, QB, MB, SMB, SB, LB, AC, constant, *VD, *LD, *AC	Byte
N	VB, IB, QB, MB, SMB, SB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC	Byte

ROR -B & ROL -B	
	<p>ROR -B & ROL -B: A instrução move o byte de entrada para os N bits da direita ou para a esquerda e coloca o resultado no byte de saída (OUT). A rotação é cíclica. Se N for maior ou igual a 8, o restante de $N / 8$ é o número de bits de rotação. Se o restante for igual a 0, a operação de rotação não é executada e o valor de SM1.0 é 1. Se a operação de rotação for executada, o bit de rotação final é copiado para o bit de Estouro (SM1.1).</p> <p>ROR -B & ROL - As operações B não são assinadas.</p>

Condições de Erro:

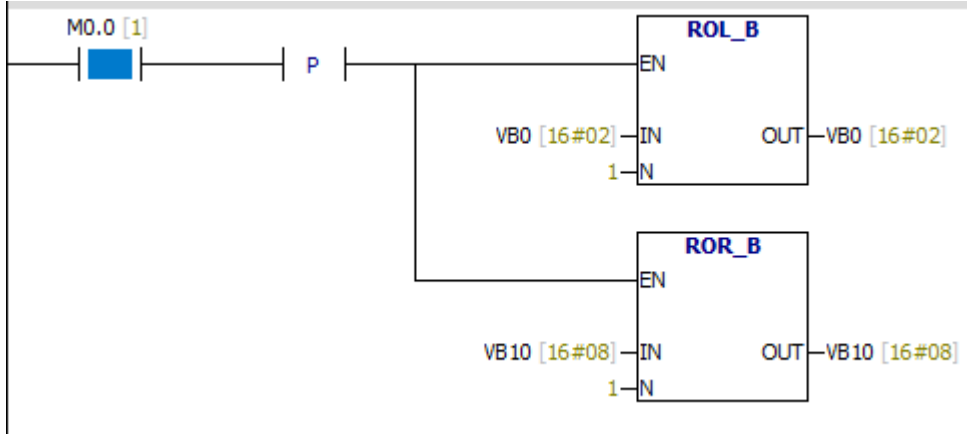
0006 Endereço Indireto

Bit memória especial:

SM1.0 Quando o valor do loop é zero, SM1.0 é definido como 1.

SM1.1 Estouro

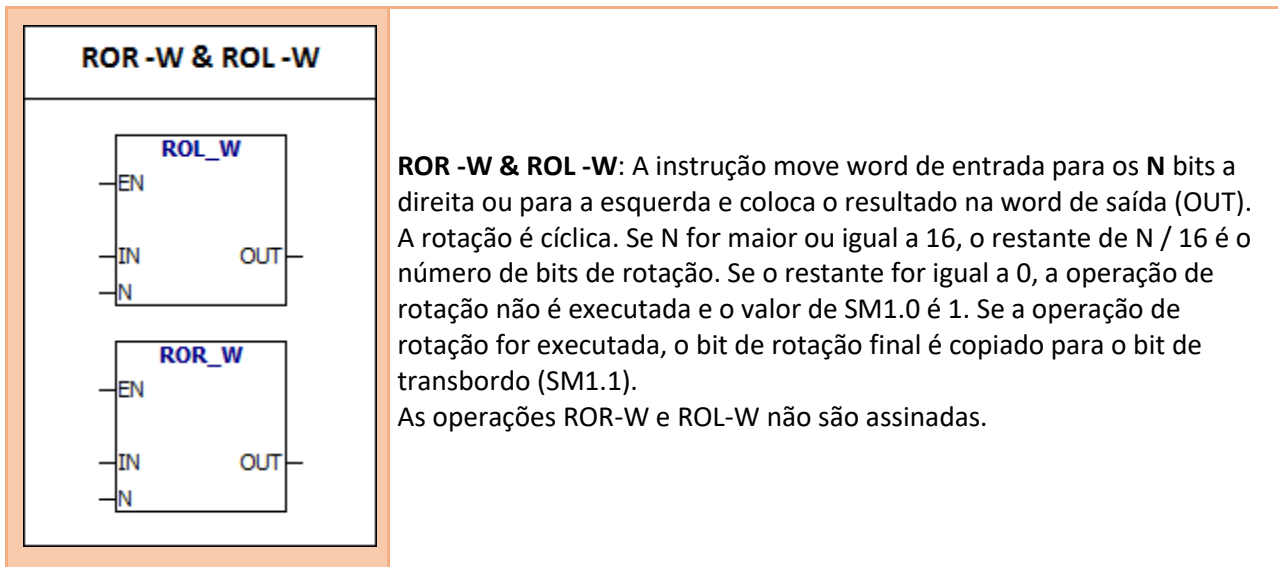
Exemplo:



Quando o valor de M0.0 é 1, VB0 move um pouco para a esquerda e VB10 move-se um pouco para a direita circularmente.

7.13.5. ROR -W & ROL -W

Entrada / Saída	Operando	Tipo de dado
IN (LAD, FBD)	VW, T, C, IW, QW, MW, SW, SMW, LW, AC, AIW, constant, *VD, *LD, *AC	Word
	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VW, T, C, IW, QW, MW, SW, SMW, LW, AC, *VD, *LD, *AC	Word



Condições de Erro:

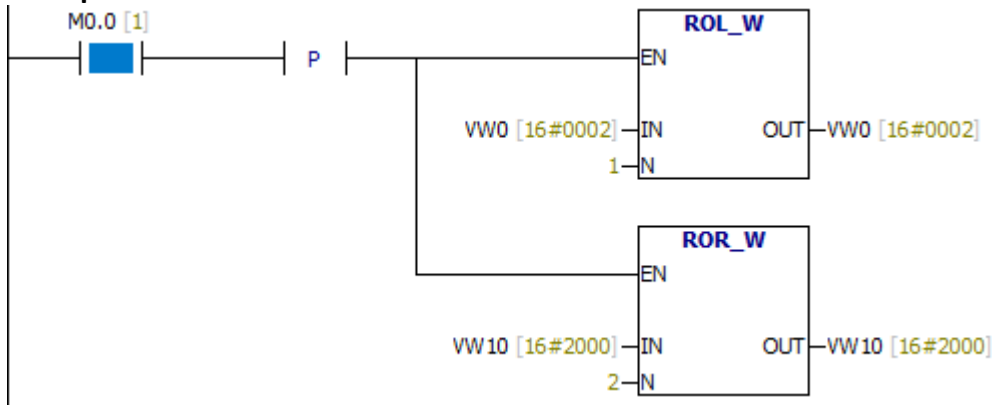
0006 Endereço Indireto

Bit memória especial:

SM1.0 Quando o valor do loop é zero, SM1.0 é definido como 1.

SM1.1 Estouro bit

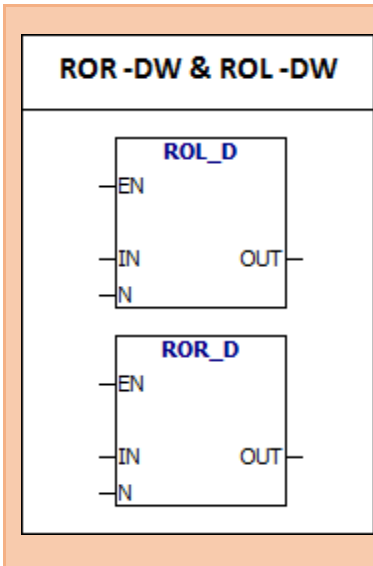
Exemplo:



Quando o valor de M0.0 é 1, VW0 move um pouco para a esquerda e VW10 move dois bits para a direita circularmente.

7.13.6. ROR -DW & ROL -DW

Entrada / Saída	Operando	Tipo de dado
IN (LAD, FBD)	VD, ID, QD, MD, SD, SMD, LD, AC, HC, constant, *VD, *LD, *AC	Double word
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *LD, *AC	Double word



ROR -DW & ROL -DW: A instrução move a double word de entrada para os n bits a direita ou para a esquerda e coloca o resultado na double word de saída (OUT). A rotação é cíclica. Se N é maior ou igual a 32, o restante de N / 32 é o número de bits de rotação. Se o restante for igual a 0, a operação de rotação não é executada e o valor de SM1.0 é 1. Se a operação de rotação for executada, o bit de rotação final é copiado para o bit de transbordo (SM1.1). ROR - As operações DW & ROL -DW não são assinadas.

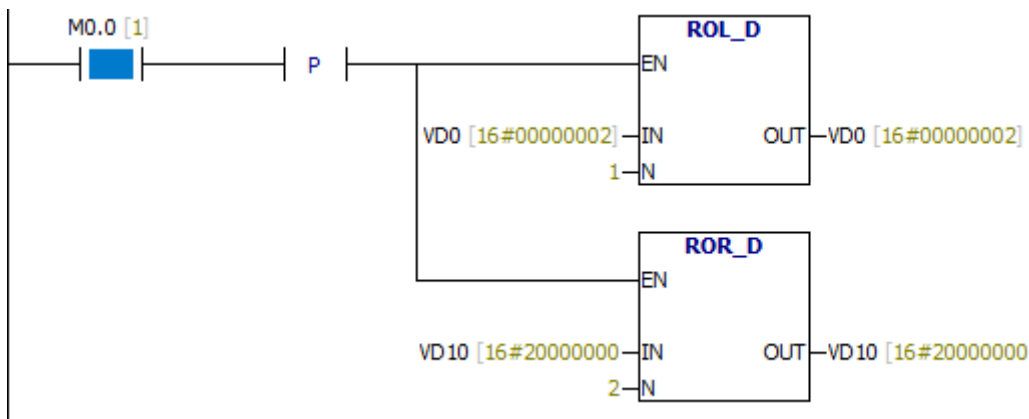
Condições de Erro:

0006 Endereço Indireto

Bit memória especial:

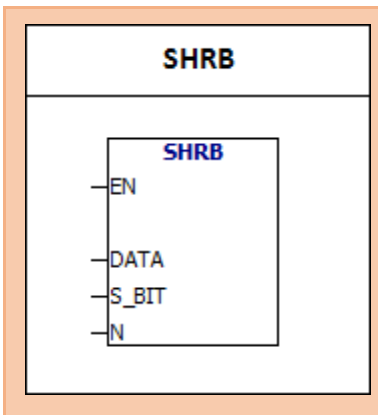
SM1.0 Quando o valor do loop é, SM1.0 é definido como 1.

Exemplo:



7.13.7. SHRB

Entrada / Saída	Operando	Tipo de dado
DATA, S_BIT	I, Q, M, SM, T, C, V, S, L	Booleana
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Booleana



A instrução **SHRB** move o valor DATA para o registro de deslocamento. S_BIT especifica o bit mais baixo do registro de deslocamento. N especifica o comprimento do registro de deslocamento e a direção de mudança (shift plus = N, shift minus = N).

O bit movido é colocado no bit de memória de estouro (SM1.1). A instrução é definida por S_BIT e N.

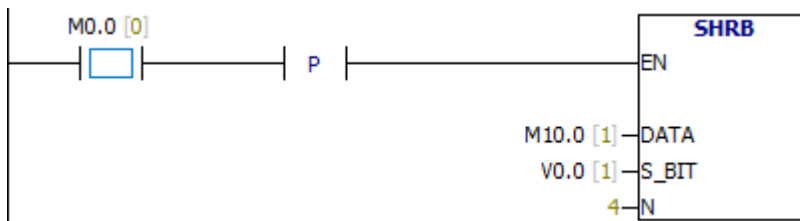
Condições de Erro:

- 0006 Endereço Indireto
- 0091 Número de operação está fora de alcance

Bit memória especial:

- SM1.1 Estouro bit

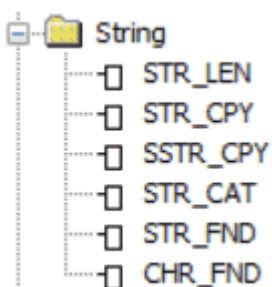
Exemplo:



Quando o valor de M0.0 é 1, o valor de M10.0 é movido para V0.0, o valor de V0.0 é movido para V0.1, o valor de V0.1 é movido para V0.2, o valor de V0.2 é movido para V0.3, o valor de V0.3 é movido para SM1.1.

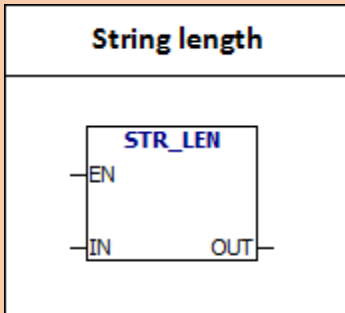
Se N for negativo, a direção de mudança é oposta.

7.14. String



7.14.1. Tamanho da String

Entrada / Saída	Operando	Tipo de dado
IN	VB, Constant string, LB, *VD, *LD, *AC	Caracter da String
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	Byte



A instrução **STR-LEN** mede o tamanho da string e envia para "OUT".

A string constante mais longa é de 126 bytes.

Condições de Erro:

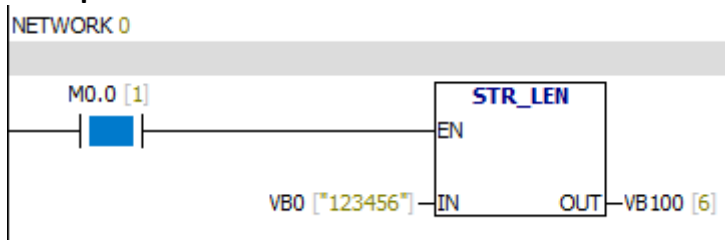
0006 Endereço Indireto

0091 Alcance de operação

Tipo de dado string constante no formato ASCII :

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

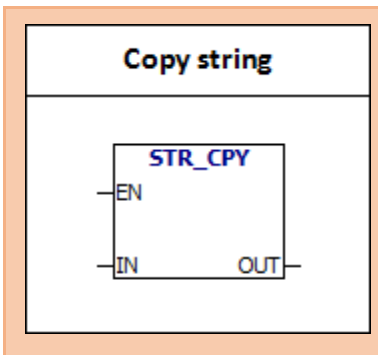
Exemplo:



Quando a string é "123456", o comprimento da string é 6.

7.14.2. Copiar String

Entrada / Saída	Operando	Tipo de dado
IN	VB, Constant string, LB, *VD, *LD, *AC	Caracter da String
OUT	VB, *VD, LB, *LD, *AC	Caracter da String



A instrução **STR-CPY** copia a string "IN" e enviar para "OUT".
A string constante mais londa é de 126 bytes

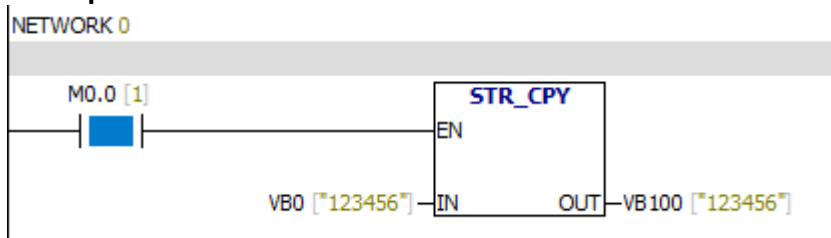
Condições de Erro:

- 0006 Endereço Indireto
- 0091 Alcance de operação

Tipo de dado string constante no formato ASCII :

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

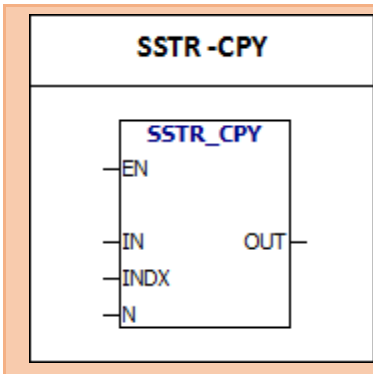
Exemplo:



Quando M0.0 é 1, a string que começa com o VB0 é copiada para a string que começa com VB100. O armazenamento VB100 é um número inteiro 6, o armazenamento VB101 é o caractere "1", o armazenamento VB102 é o caractere "2", o armazenamento VB103 é "3" O armazenamento VB104 é "4", o armazenamento VB105 é "5", o armazenamento VB106 é "6".

7.14.3. SSTR-CPY

Entrada / Saída	Operando	Tipo de dado
Iput	VB, Constant string, LB, *VD, *LD, *AC	string
INDX, N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	byte
OUT	VB, *VD, LB, *LD, *AC	string



SSTR-CPY: copia uma parte da sequência de entrada para a string OUT. Se o valor de INDX for N, copia a string a partir do N-ésimo caractere. O comprimento da string de cópia é N. A string constante mais longa é de 126 bytes.

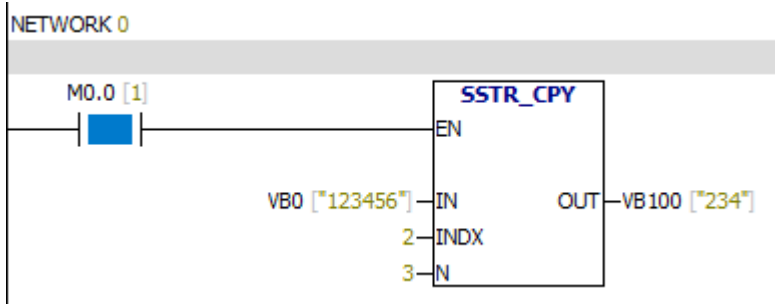
Condições de Erro:

- 0006 Endereço Indireto
- 0091 Alcance de operação
- 009B Índice é ilegal

Tipo de dado string constante no formato ASCII :

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

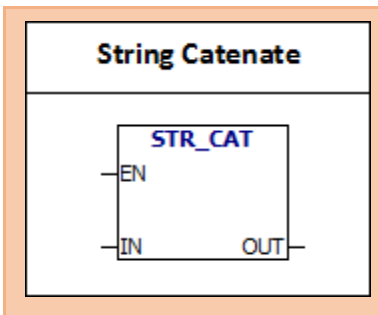
Exemplo:



Copia a string VB0. Copia a string a partir do segundo caractere. O comprimento da sequência de cópia é 3. O resultado é colocado no VB100.

7.14.4. Concatenar String

Entrada / Saída	Operando	Tipo de dado
IN	VB, Constant string, LB, *VD, *LD, *AC	String
OUT	VB, LB, *VD, *LD, *AC	String



STR -CAT: Adiciona a string especificada pelo IN para string especificada pelo OUT.

A string constante mais longa é de 126 bytes.

Condições de Erro:

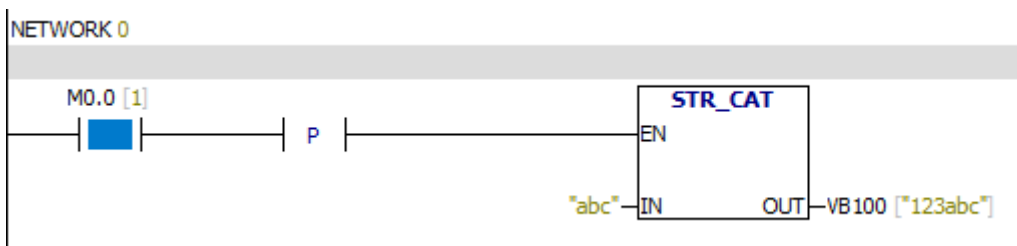
0006 Endereço Indireto

0091 Alcance de operação

Tipo de dado string constante no formato ASCII :

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

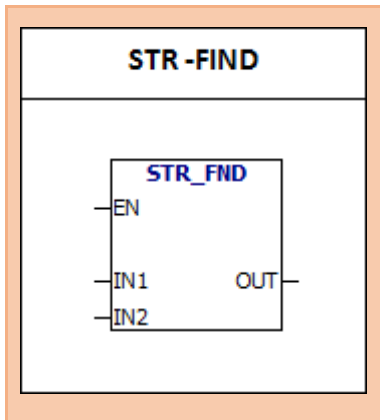
Exemplo:



A string VB100 é "123". Depois de usar a instrução STR -CAT, a seqüência VB100 é "123abc".

7.14.5. STR-FIND

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VB, constant string, LB, *VD, *LD, *AC	String
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	Byte



STR -FIND: A instrução busca a string IN2 na string IN1. A pesquisa começa a partir da posição de início de saída. Se você encontrar uma string que é a mesma que a string IN2, a primeira posição de caractere da string de caracteres é gravada em OUT. Se você não encontrar o IN2 em IN1, o OUT é definido como 0. O comprimento mais longo de uma única string constante é 126 bytes. Os comprimentos mais longos de duas strings constantes são de 240 bytes.

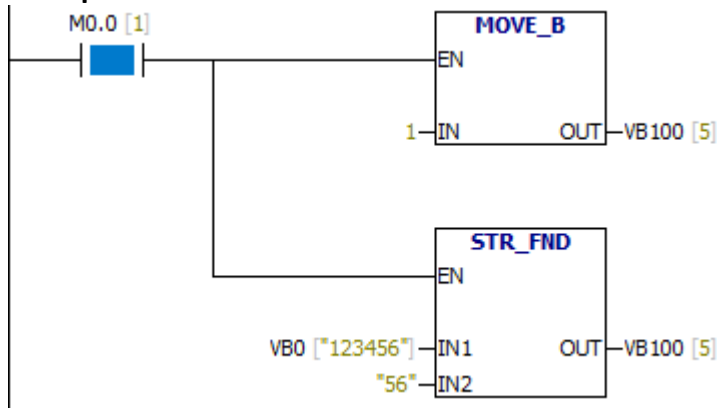
Condições de Erro:

0006 Endereço Indireto
0091 Alcance de operação
009B Índice Inválido

Tipo de dado string constante no formato ASCII :

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, A string deve começar e terminar com aspas duplas ("string constant").

Exemplo :



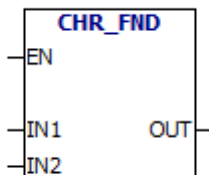
Atenção:

Você precisa mover 1 para VB100, então a instrução STR-FND pode pesquisar a string "IN2" do primeiro caractere de "IN1".

7.14.6. Procura o primeiro caractere na string

Entrada / Saída	Operando	Tipo de dado
IN1, IN2	VB, constant string, LB, *VD, *LD, *AC	String
OUT	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *LD, *AC	Byte

Lê o primeiro caractere da String



CHR -FIND: A instrução busca o mesmo caractere na string IN2 na string IN1. A pesquisa começa a partir da posição de início OUT. Se um caractere de correspondência for encontrado, a posição do caractere é escrita para OUT. Se um caractere de correspondência não for encontrado, o OUT é Definido como 0. O comprimento mais longo de uma única string constante é 126 bytes. Os comprimentos abrangentes mais longos de duas strings constantes são de 240 bytes.

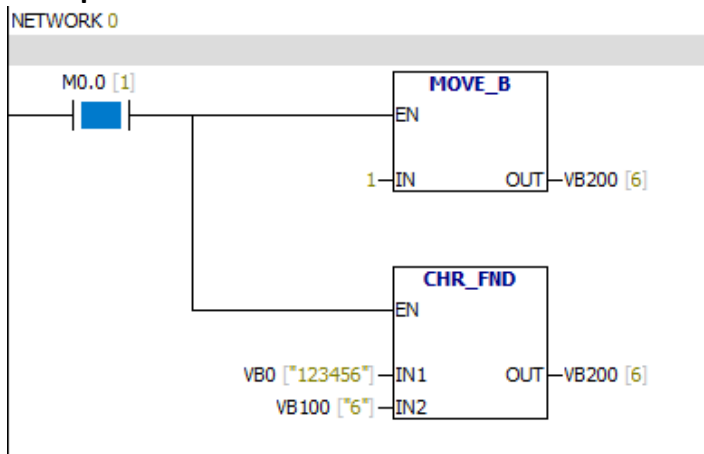
Condições de Erro:

- 0006 Endereço Indireto
- 0091 Alcance de operação
- 009B Índice inválido

Tipo de dado string constante no formato ASCII :

String é uma série de caracteres, cada caractere é armazenado como um byte. O primeiro byte de uma string define o comprimento da string, ou seja, o número de caracteres. Se uma string constante for inserida diretamente no editor do programa ou bloco de dados, a string deve começar e terminar com aspas duplas ("string constant").

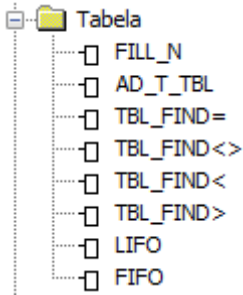
Exemplo :



Atenção:

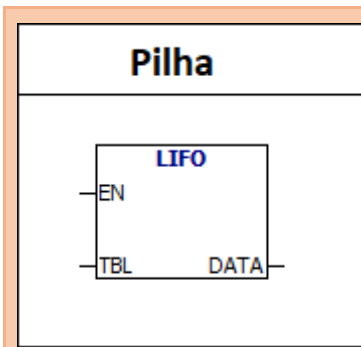
Você precisa mover 1 para VB200, então a instrução CHR-FND pode pesquisar os caracteres de "IN2" em "IN1".

7.15. Tabela



7.15.1. Último a entrar é o primeiro a sair

Entrada / Saída		Tipo de dado
TBL	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC	Word
DATA	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, AQW, *VD, *LD, *AC	Integer



LIFO: A instrução move a entrada mais recente (ou última) na tabela para o endereço de memória de saída. Remova a última entrada na tabela (TBL) e mova o valor para a localização especificada por DATA. Cada vez que a instrução é executada, o número de entradas na tabela reduz 1.

Condições de Erro:

- 0006 Endereço Indireto
- 0091 Alcance de operação

Bit memória especial:

- SM1.5 Lista vazia

Formato de Tabela

VW200	Número máximo de entrada
VW202	Contagem de entrada
W204	Data 0
VW206	Data 1
VW208	Data 2
.....

Exemplo:

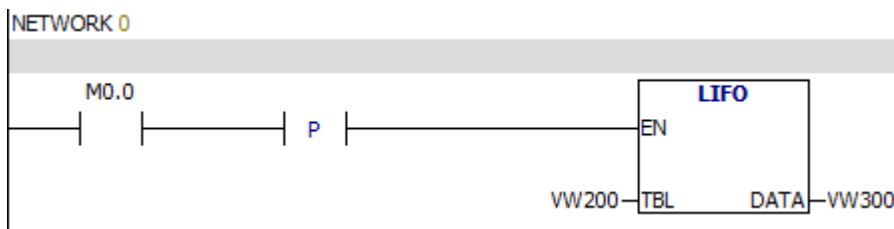
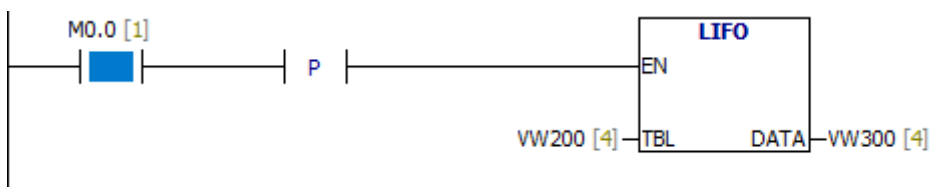


Tabela de monitoramento:

	Adress	Data Type	Val
✓	VW200	INT	4
✓	VW202	INT	4
✓	VW204	INT	1
✓	VW206	INT	2
✓	VW208	INT	3
✓	VW210	INT	4

Análise:

Quando o valor de M0.0 for igual a 1, a última entrada da tabela será excluída e o valor da última entrada da tabela será movido para "VW300".



VW200	INT	4
VW202	INT	3
VW204	INT	1
VW206	INT	2
VW208	INT	3
VW210	INT	4

Quando o valor de M0.0 é igual a 1:

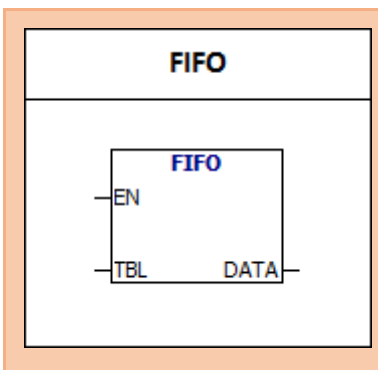
VW202=3

VW210 inválido

VW300=4

7.15.2. Primeiro a entrar é o primeiro a sair

Entrada / Saída	Operando	Tipo de dado
TBL	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC	word
DATA	VW, IW, QW, MW, SW, SMW, LW, AC, T, C, AQW, *VD, *LD, *AC	integer



FIFO: Remove a primeira entrada na tabela (TBL) e move o valor para a localização especificada por DATA. Todas as outras entradas na tabela movem uma posição para cima. Cada vez que a instrução é executada, o número de entradas na tabela reduz 1.

Condições de Erro:

0091 Alcance de operação

Bit memória especial:

SM1.5 Lista vazia

Formato de Tabela

VW200	Número máximo de entrada
VW202	Contagem de entrada
W204	Data 0
VW206	Data 1
VW208	Data 2
.....

Exemplo:

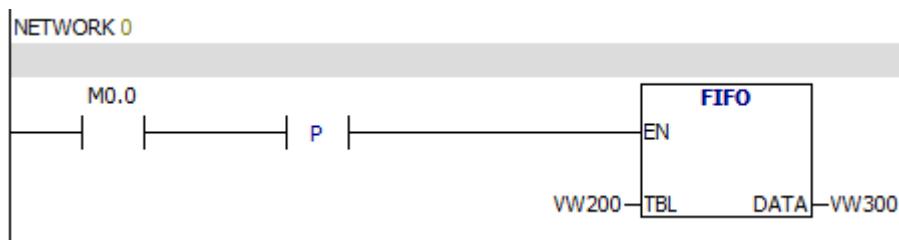
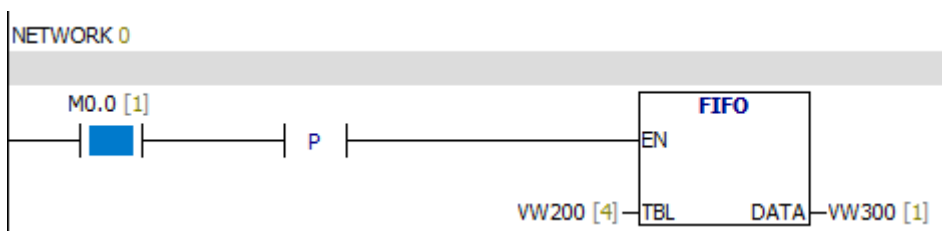


Tabela de monitoramento:

	Adress	Data Type	Value
✓	VW200	INT	4
✓	VW202	INT	4
✓	VW204	INT	1
✓	VW206	INT	2
✓	VW208	INT	3
✓	VW210	INT	4

Analyze

Quando o valor de M0.0 é igual a 1, a primeira entrada da tabela será excluída e o valor da primeira entrada da tabela será movido para "VW300".



VW200	INT	4
VW202	INT	3
VW204	INT	2
VW206	INT	3
VW208	INT	4
VW210	INT	4

Quando o valor de M0.0 é igual a 1:

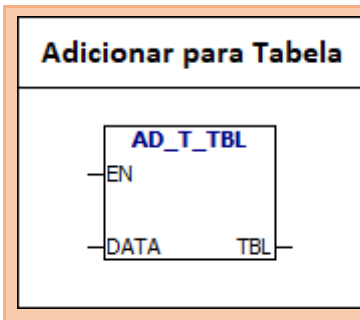
VW202=3

VW210 inválido

VW300=1

7.15.3. Adicionar à tabela

Entrada / Saída	Operando	Tipo de dado
DATA	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constant, *VD, *LD, *AC	Integer
TBL	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD *AC	Word



AD -T- TBL: A instrução adiciona a word (DATA) à tabela (TBL). O primeiro valor na tabela é o comprimento máximo da tabela. O segundo valor é a contagem de entrada (EC), ele especifica o número de entradas na tabela. Cada vez que você adiciona novos dados à tabela, o número de entradas adiciona 1. Tabela pode conter até 100 entradas, não incluindo a primeira entrada e a segunda entrada.

Condições de Erro:

0006 Endereço Indireto

0091 Alcance de operação

Bit memória especial:

SM1.4 Table Estouro

Exemplo:

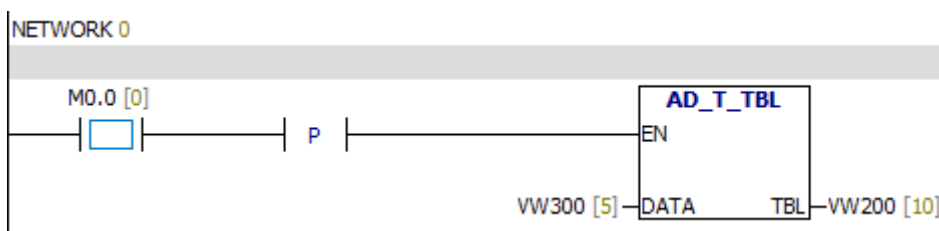


Tabela de monitoramento:

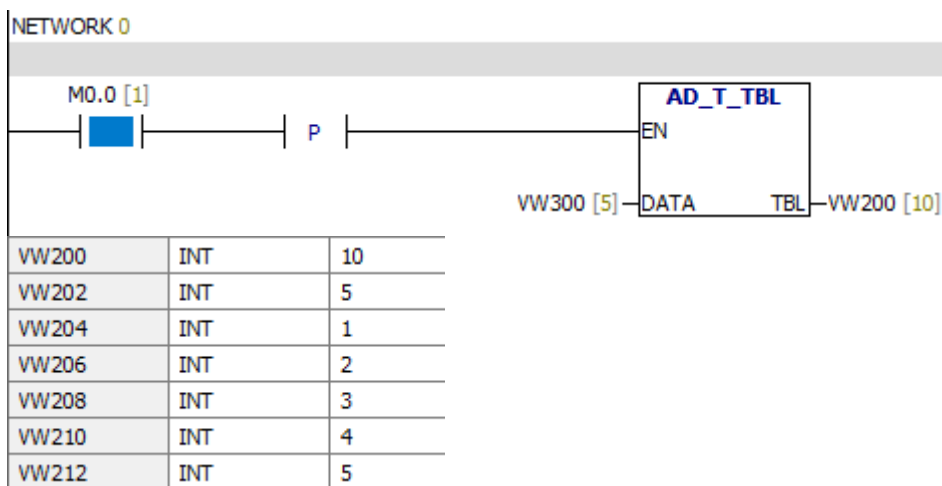
	Adress	Data Type	Value
✓	VW200	INT	10
✓	VW202	INT	4
✓	VW204	INT	1
✓	VW206	INT	2
✓	VW208	INT	3
✓	VW210	INT	4

Quando o valor de M0.0 é igual a 1:

O valor de VW202 + 1

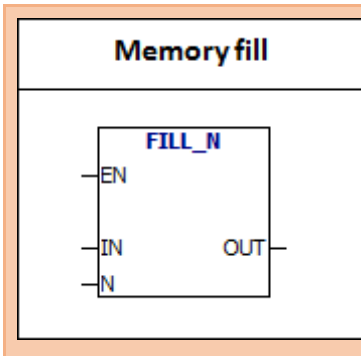
A tabela terá uma nova entrada

O valor da nova entrada é igual ao valor do VW300.



7.15.4. Preencher memória

Entrada / Saída	Operando	Tipo de dado
IN	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, AC, constant, *VD, *LD, *AC	Integer
N	VB, IB, QB, MB, SB, SMB, LB, AC, constant, *VD, *LD, *AC	Byte
OUT	VW, IW, QW, MW, SW, SMW, LW, T, C, AQW, *VD, *LD, *AC	Integer



FILL-N: O valor de entrada em "IN" é escrito nas words contínuas "OUT" N.

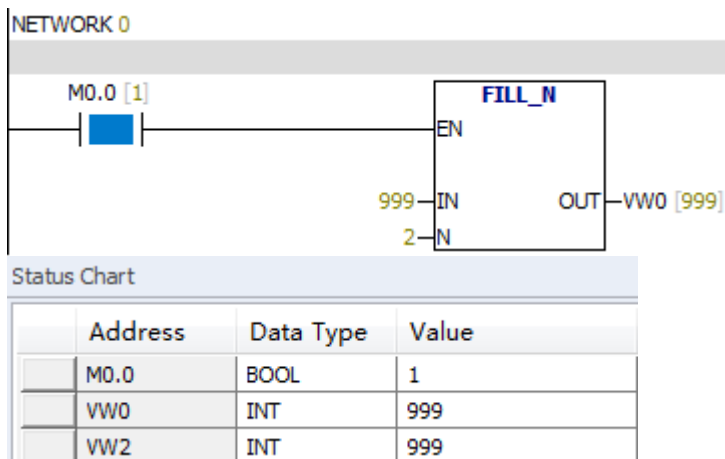
O intervalo de N é de 1 a 255.

Condições de Erro:

0006 Endereço Indireto

0091 Alcance de operação

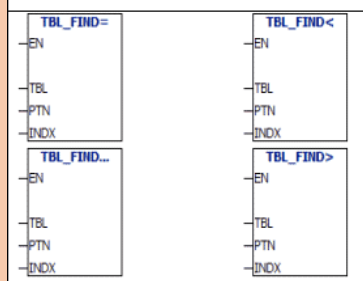
Exemplo:



7.15.5. Tabela de busca

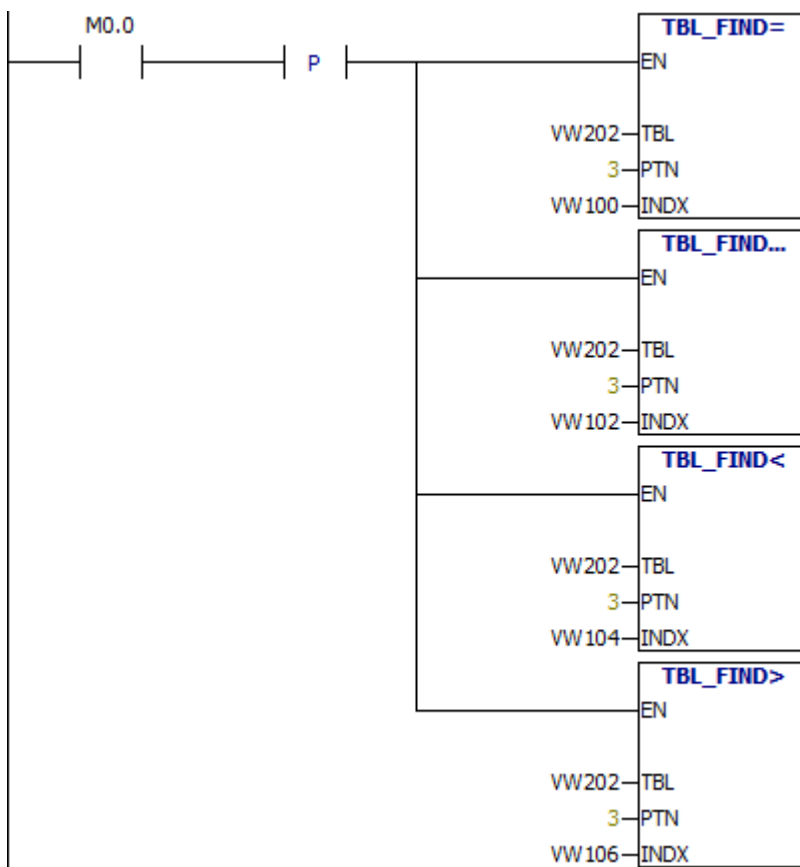
Entrada / Saída	Operando	Tipo de dado
TBL	VW, IW, QW, MW, SW, SMW, LW, T, C, *VD, *LD, *AC	Word
PTN	VW, IW, QW, MW, SW, SMW, AIW, LW, T, C, AC, constant, *VD, *LD, *AC	Integer
INDX	VW, IW, QW, MW, SW, SMW, LW, T, C, AC, *VD, *LD, *AC	Word

Pesquisa de tabela



Instruções de pesquisa de tabela: Uma instrução procura os mesmos dados que "PTN" na tabela. "Tabela de busca" inicia uma inserção a entrada especificada pelo INDX. Se uma entrada correspondente para encontrada, o INDX aponta para a entrada na tabela. Para encontrar uma entrada coincidente do Próximo, você deve adicionar 1 ao INDX antes de usar uma instrução "Tabela de busca". Se a entrada não for encontrada, o valor de INDX será igual ao número de entradas.

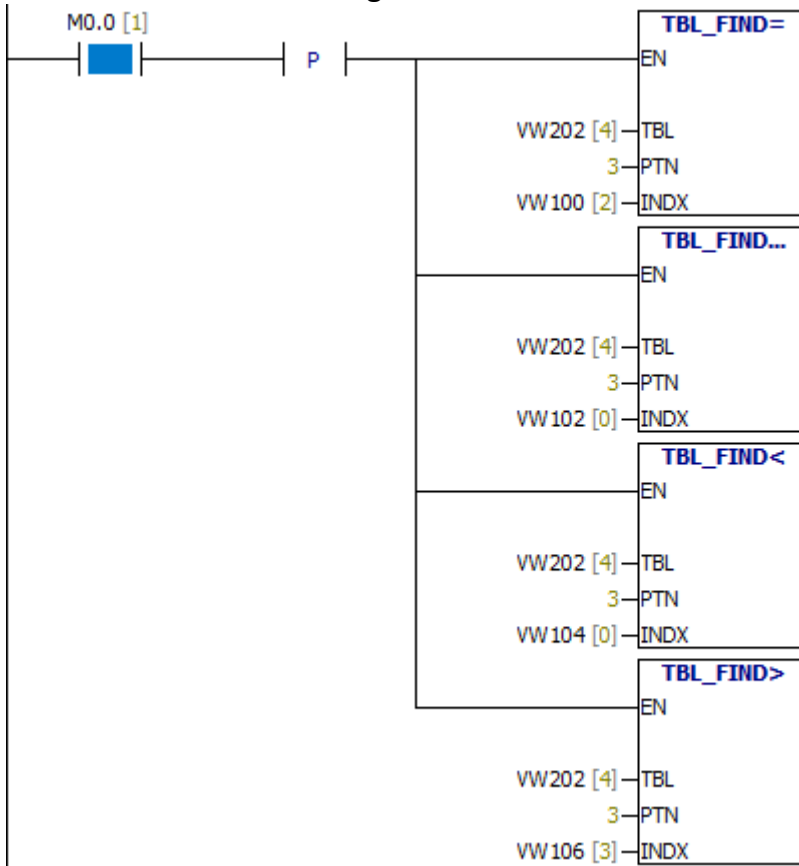
Exemplo:



Bloco de dados:

	Adress	Data Type	Value
✓	VW200	INT	10
✓	VW202	INT	4
✓	VW204	INT	1
✓	VW206	INT	2
✓	VW208	INT	3
✓	VW210	INT	4

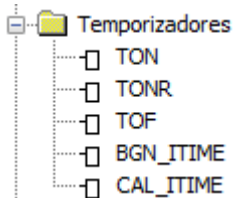
Quando o valor de M0.0 é igual a 1:



O formato da tabela do "Table-Find" começa com a contagem de entradas. Não possui "número máximo de entradas":

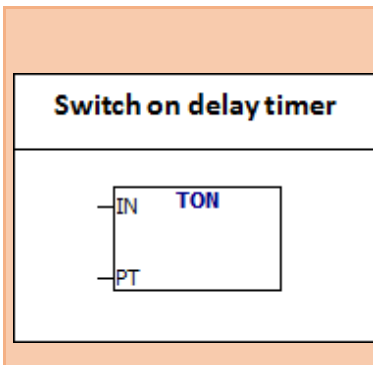
VW202	Contagem de entrada
VW204	Data 0
VW206	Data 1
VW208	Data 2
VW210	Data 3

7.16. Temporizadores



7.16.1. TON

Entrada / Saída	Operando	Tipo de dado
Txxx	Constant (T0 -T255)	Word
IN (LAD)	Enable bit	Booleana
IN (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
PT	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constant, *VD, *LD, *AC Integer	Integer



TON: Quando o valor da entrada "IN" é igual a 1, inicia o tempo do temporizador. O valor atual do Txxx é o tempo atual (um múltiplo da base de tempo). Quando o valor atual do temporizador é igual ao tempo predefinido (PT), o valor do bit do temporizador é 1. Quando o valor da entrada "IN" é igual a 0, o valor atual do temporizador é apagado. Os temporizadores TONR e TOF têm três tipos de resoluções. Cada valor atual é um múltiplo Da base de tempo. Para o Exemplo, o número 50 no temporizador de 10 milissegundos é 500 milissegundos.

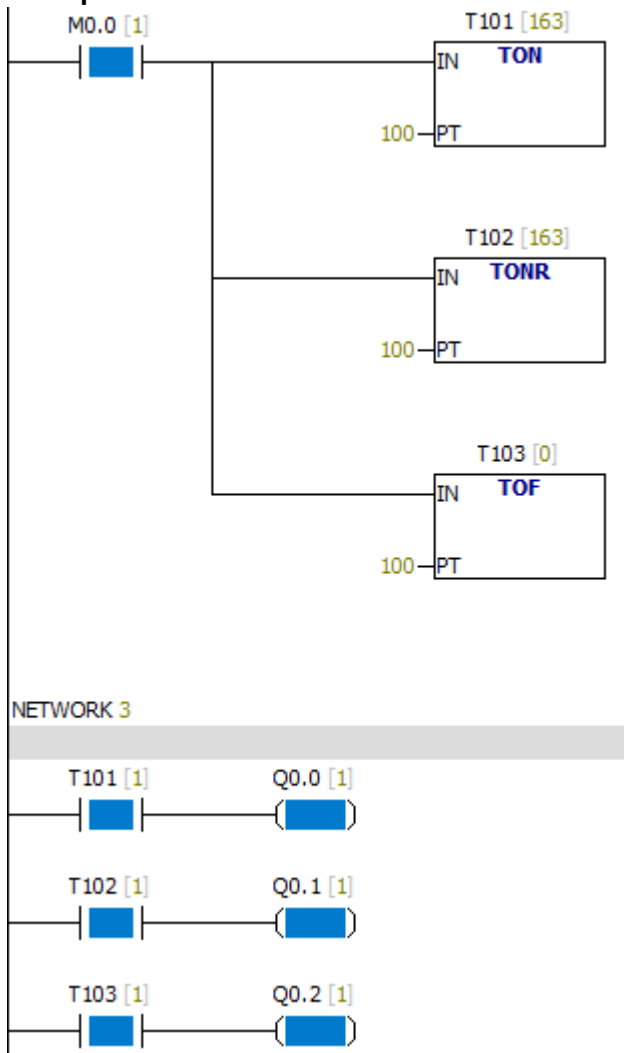
Intervalo do temporizador:

Número do temporizador	Base de tempo (ms)	Intervalo de tempo (s)
T0	1	65.535
T1~T4	10	655.35
T5~T31	100	6553.5
T32	1	65.535
T33~T36	10	655.35
T37~T63	100	6553.5
T64	1	65.535
T65~T68	10	655.35
T69~T95	100	6553.5
T96	1	65535
T97~T100	10	655.35
T101~T127	100	6553.5

Atenção:

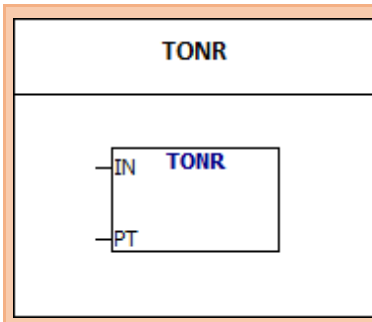
1. O valor de cada temporizador TXXX é diferente.
2. A resolução do temporizador depende da base de tempo selecionada na tabela acima. Por exemplo, utilizando o contato T1 com o valor 10, totaliza um tempo de 100ms.

Exemplo:



7.16.2. TONR

Entrada / Saída	Operando	Tipo de dado
Txxx	Constant(T0—T255)	word
IN (LAD)	Enable bit	Booleana
IN (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
PT	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constant, *VD, *LD, *AC	Integer



TONR: Quando o valor da entrada "IN" é igual a 1, inicia o tempo do temporizador. O valor atual do Txxx é o tempo atual (um múltiplo da base de tempo). Quando o valor atual do temporizador é igual ao tempo predefinido (PT), o valor do bit do temporizador é 1. Quando o valor da entrada "IN" é igual a 0, se o valor atual do temporizador for menor que o valor predefinido, o valor atual do temporizador é mantido. Caso contrário, o valor do temporizador é apagado.

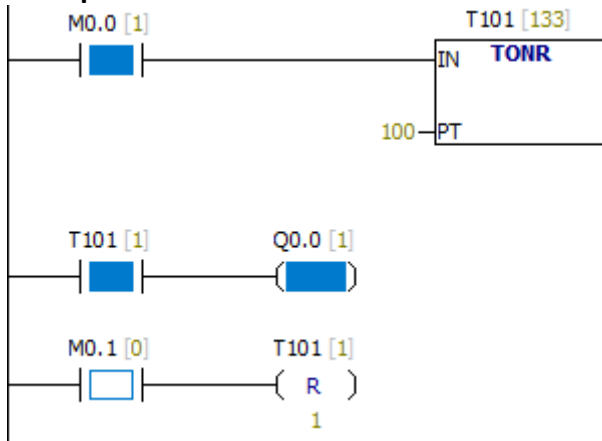
Notas:

Você pode usar TONR para acumular vários intervalos de tempo.

Você pode usar a instrução "recuperação" (R) para recuperar qualquer temporizador.

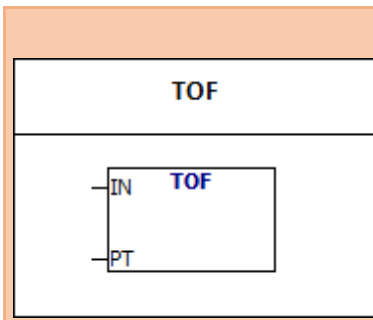
Você só pode usar a instrução "recuperação" para recuperar o temporizador TONE.

Exemplo:



7.16.3. TOF

Entrada / Saída	Operando	Tipo de dado
Txxx	Constant(T0—T255)	Word
IN (LAD)	Enable bit	Booleana
IN (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Booleana
PT	VW, IW, QW, MW, SW, SMW, LW, AIW, T, C, AC, constant, *VD, *LD, *AC	Integer



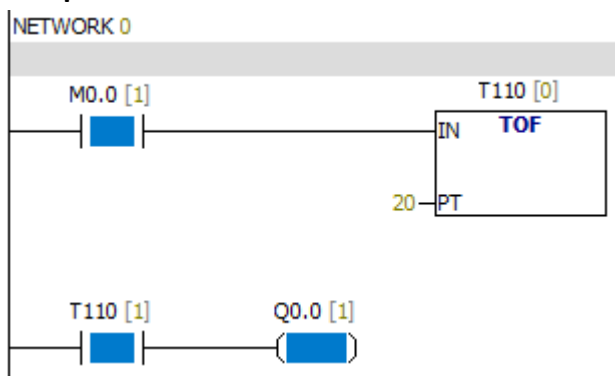
TOF: Quando a entrada é fechada, a saída será fechada por um período de tempo. Quando o valor de IN for 1, o bit do temporizador é 1 imediatamente e o valor atual do temporizador é definido como 0. Quando o valor de IN é 0, O temporizador inicia o tempo. Quando o valor atual é igual ao valor predefinido, o bit do temporizador é 0.

Notas:

O valor de cada temporizador TXXX é diferente.

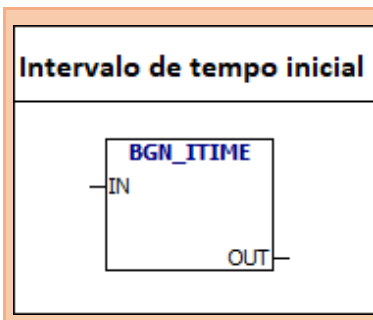
Você pode usar a instrução "recuperação" (R) para recuperar o temporizador TOF.

Exemplo:



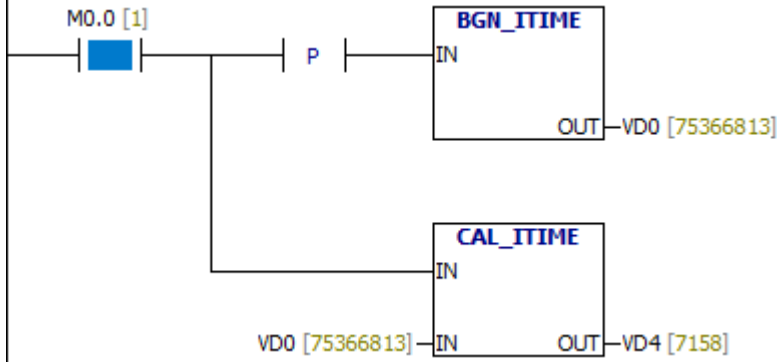
7.16.4. Intervalo de tempo inicial

Entrada / Saída	Operando	Tipo de dado
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Double word



Lê o valor atual do contador incorporado de 1 ms e o armazena no OUT.

Exemplo:



O valor de VD4 é o tempo de condução de M0.0

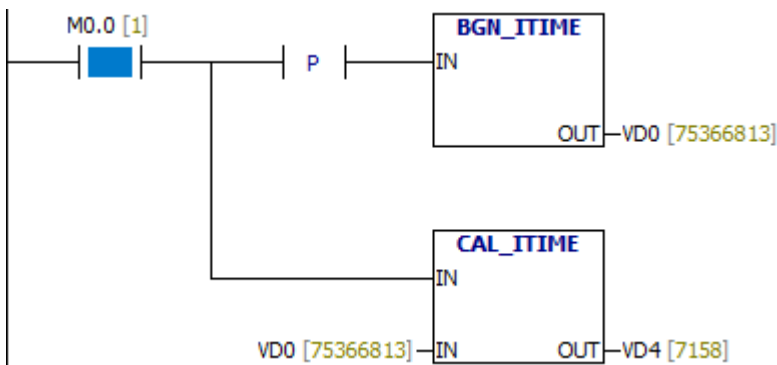
7.16.5. Tempo de intervalo de cálculo

Entrada / Saída	Operando	Tipo de dado
IN	VD, ID, QD, MD, SMD, SD, LD, HC, AC, *VD, *LD, *AC	Double word
OUT	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC	Double word

Intervalo de tempo inicial

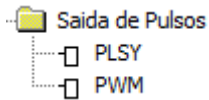
Calcula a diferença horária entre a hora atual e a hora fornecida pelo IN e armazena a diferença horária no OUT.

Exemplo:



O valor de VD4 é o tempo de condução de M0.0

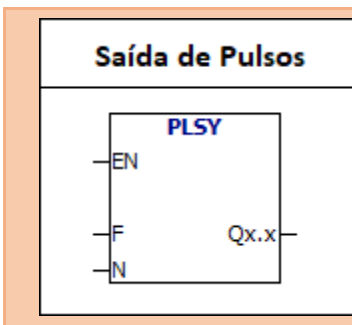
7.17. Trem de Pulsos



[Como chamar o arquivo de ajuda](#)

7.17.1. Saída de Pulsos

Entrada / Saída	Operando	Tipo de dado
F	ID, QD, AID, AQD, MD, VD, HC, SMD, LD, *MD, *VD, *LD	Double integer
N	ID, QD, AID, AQD, MD, VD, HC, SMD, LD, *MD, *VD, *LD	Double integer
OUT	QX.X	Bit

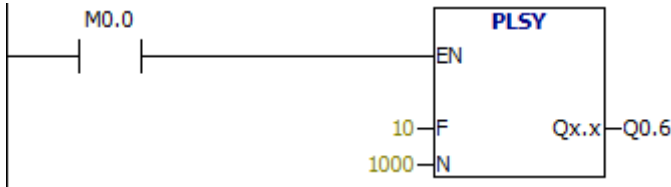


PLSY: Quando o valor do bit de habilitado é 1, a instrução emite N pulsos. A frequência de pulso é F.

Instruções PLSY:

1. A faixa de frequência de F é 10 ~ 40K (Hz). Os modelos diferentes têm diferentes faixas de frequência. Por favor, defina a frequência de acordo com o modelo específico. A frequência F pode ser alterada no processo de transmissão de pulso, a frequência de pulso de envio também é alterada.
2. O intervalo de N é 0 ~ 2147483647. Se N é 0, o número de pulsos é ignorado. Quando n é igual a 0 eo bit de habilitação é 1, a instrução PLSY enviará pulsos incessantemente.
Quando o pulso está sendo enviado, mudar o valor de N não funciona. N mudanças entrarão em vigor após o próximo pulso.
3. Se o valor do bit de habilitação for 0, o pulso irá parar de enviar. Quando o bit de habilitação é alterado de 0 para 1, a instrução PLSY envia novos impulsos e ignora os pulsos interrompidos antes.
4. O índice de taxa de transmissão de pulsos é de 50% ON, 50% OFF. A transmissão do pulso é completamente processada pela interrupção do hardware, que não é afetada pelo período de varredura.

Exemplo:



10 pulsos por segundo, um total de 1000 pulsos enviados.

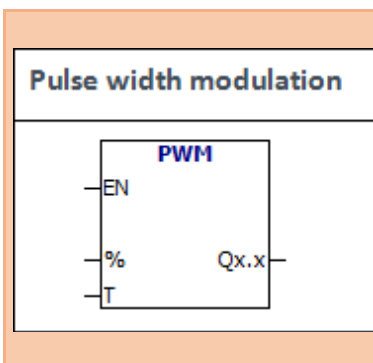
Atenção:

O ponto de saída deve ser o ponto de alta velocidade.

Para diferentes CLPs, os endereços dos pontos de saída rápida podem ser diferentes.

7.17.2. Módulo de largura de pulso

Entrada / Saída	Operando	Tipo de dado
%	IW, QW, AIW, AQW, MW, VW, T, C, SMW, LW, *MD, *VD, *LD	Double word
T	IW, QW, AIW, AQW, MW, VW, T, C, SMW, LW, *MD, *VD, *LD	Double word
OUT	Q	Bit



A instrução, Módulo de Largura de Pulso (PWM) inicializa o hardware PWM e envia pulsos de alta velocidade.

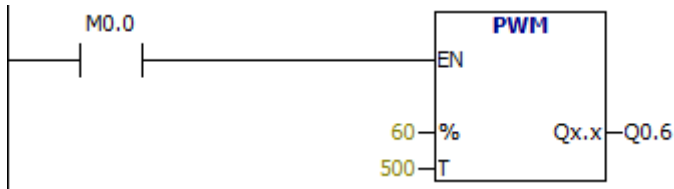
O valor de entrada de "%" = **tempo / período de condução**.

O valor de entrada de "T" é o período do pulso.

Descrição do PWM:

1. A unidade de T é 1 ms.
2. Se o valor de entrada "%" for 0, a instrução não emitirá o pulso. Se o valor de entrada de "%" for igual a 100, o valor do pulso de saída é sempre 1.
3. Quando o pulso está sendo enviado, você pode alterar o valor de "%" e o período do pulso. Então, o valor de "%" e o período de pulso vão mudar.
4. Se o valor do bit de habilitação for 0, o envio de pulsos serem interrompidos. Quando o bit de habilitação for alterado de 0 para 1, a instrução PWM reinicia o envio de pulsos.

Exemplo:



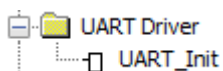
O período do pulso é 500ms, o tempo de condução é de 300 ms.

Atenção:

O ponto de saída deve ser o ponto de saída de alta velocidade.

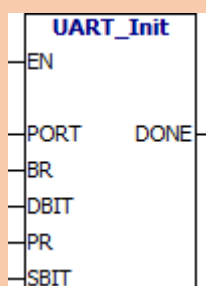
Para diferentes CLP, os endereços dos pontos de saída de alta velocidade podem ser diferentes.

7.18. Driver UART



[Como chamar o arquivo de ajuda](#)

7.18.1. UART-Init



UART_Init a instrução é usada para inicializar a porta serial.

EN : Se o valor de entrada for igual a 1, a instrução inicializará a porta serial.

PORT : número da porta, 0 à 2.

Br: Taxa de transmissão da porta serial.

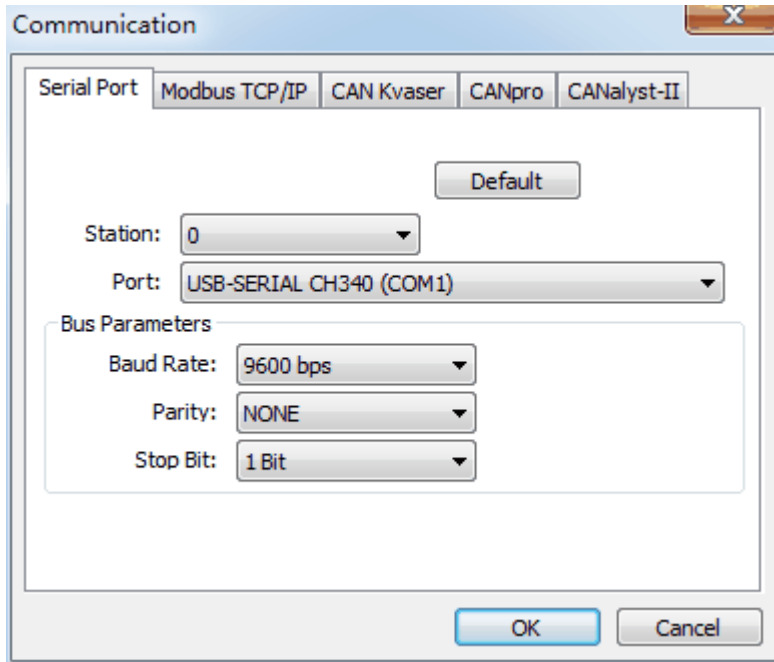
DBIT : Número da serial data bit.

PR : Bit de verificação da porta Serial, 0 = Sem paridade, 1 = ímpar, 2 = verificação de paridade.

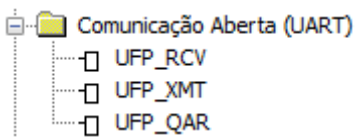
SBIT : bit de parada.

DONE : sucesso = 1, falha = 0.

Você também pode definir esses parâmetros no software de programação. Como mostrado a imagem abaixo:

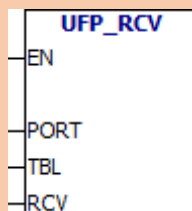


7.19. Comunicação Aberta (UART)



[Como chamar o arquivo de ajuda](#)

7.19.1. UFP-RCV



UFP_RCV: Instrução que recebe dados

TBL: Tabela de configuração, se a entrada for MB200.

MB200 é o byte de configuração:

(Instrução de saída) M200.0 Preparação de Comunicação

(Instrução de saída) M200.1 Comunicação completa

(Instrução de saída) M200.2 Erro de Comunicação

(Instrução de entrada) M200.3 Envia cheque CRC

(Instrução de entrada) M200.4 Envia cheque CRC

(Instrução de entrada) M200.5 Recebe cheque CRC

(Instrução de entrada) M200.6 Recebe cheque CRC

RCV: Recebe os dados, se a entrada for MB400:

Instrução de entrada) MW400: Recebe dados FIFO tamanho do buffer (unidade de byte)

Instrução de saída) MW402: Tamanho dos dados recebidos (em bytes)

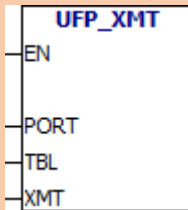
Instrução de saída) MB404 ~ ... recebe dado.

UFP_RCV、UFP_XMT、UFP_QAR número de erro:

- ✂ 1 Porta não existe
- ✂ 2 Porta não está habilitada
- ✂ 3 A fila de tarefas de comunicação está cheia
- ✂ 4 Erro de tabela
- ✂ 5 Erro de dados enviados
- ✂ 6 Tempo esgotado
- ✂ 7 Erro de dados recebidos
- ✂ 8 Erro de verificação de dados de recepção

[Exemplo](#)

7.19.2. UFP-XMT



UFP_XMT: Instrução que envia dados

PORT: Porta de comunicação

TBL: Tabela de configuração, se a entrada for MB200.

(Instrução de saída) M200.0 Preparação de Comunicação

(Instrução de saída) M200.1 Comunicação completa

(Instrução de saída) M200.2 Erro de Comunicação

(Instrução de entrada) M200.3 Envia cheque CRC

(Instrução de entrada) M200.4 Envia cheque CRC

(Instrução de entrada) M200.5 Recebe cheque CRC

(Instrução de entrada) M200.6 Recebe cheque CRC

(Instrução de saída) MB201 Número de erro: 0 indica que não há erro
XMT Envia os dados, se a entrada for MB400:

(Instrução de entrada) MW400 Envia dados FIFO tamanho do buffer
(unidade de byte)

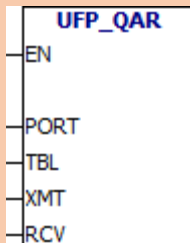
(Instrução de entrada) MW402 Tamanho dos dados enviado (em
bytes)

(Instrução de entrada) MB404 ~ ... Envia dados.

- ✎ 1 Porta não existe
- ✎ 2 Porta não está habilitada
- ✎ 3 A fila de tarefas de comunicação está cheia
- ✎ 4 Erro de tabela
- ✎ 5 Erro de dados enviados
- ✎ 6 Tempo esgotado
- ✎ 7 Erro de dados recebidos
- ✎ 8 Erro de verificação de dados de recepção

[Exemplo](#)

7.19.3. UFP-QAR



UFP_QAR: Instrução que envia e recebe dados

PORT: Porta de comunicação

TBL: Tabela de configuração, se a entrada for MB200. MB200 é o byte de configuração:

(Instrução de saída) M200.0 Preparação de Comunicação

(Instrução de saída) M200.1 Comunicação completa

(Instrução de saída) M200.2 Erro de Comunicação

(Instrução de entrada) M200.3 Envia cheque CRC

(Instrução de entrada) M200.4 Envia cheque CRC

(Instrução de entrada) M200.5 Recebe cheque CRC

(Instrução de entrada) M200.6 Recebe cheque CRC

(Instrução de saída) MB201 Número de erro: 0 indica que não há erros.

XMT: XMT Envia os dados, se a entrada for MB300:

(Instrução de entrada) MW400 Envia dados FIFO tamanho do buffer (unidade de byte)

(Instrução de entrada) MW402 Tamanho dos dados enviado (em bytes)

(Instrução de entrada) MB404 ~ ... Envia dados.

RCV receive data FIFO, if the input is MB400:

(Instrução de entrada) MW400: Recebe dados FIFO tamanho do buffer (unidade de byte)

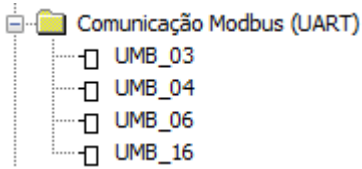
(Instrução de saída) MW402: Tamanho dos dados recebidos (em bytes)

(Instrução de saída) MB404 ~ ... recebe dado.

UFP_RCV、UFP_XMT、UFP_QAR número de erro:

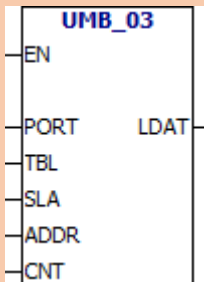
- ✎ 1 Porta não existe
- ✎ 2 Porta não está habilitada
- ✎ 3 A fila de tarefas de comunicação está cheia
- ✎ 4 Erro de tabela
- ✎ 5 Erro de dados enviados
- ✎ 6 Tempo esgotado
- ✎ 7 Erro de dados recebidos
- ✎ 8 Erro de verificação de dados de recepção

7.20. Modbus (UART)



[Como chamar o arquivo de ajuda](#)

7.20.1. UMB-03



Lê mais de um registro retentivo.

EN: habilita ou desabilita

TBL: Tabela de configuração, se a entrada for MB200:

MB200 é a word de configuração:

(Instrução de saída) M200.0 Comunicação enfileirada.

(Instrução de saída) M200.1 Comunicação completa

(Instrução de saída) M200.2 Erro de comunicação

(Instrução de saída) MB201 Número de erro: 0 indica que não há erros

SLA: MODBUS Endereço de escravo

ADDR: Ponto de ajuste de registrador retentivo (Ajuste de 4x)

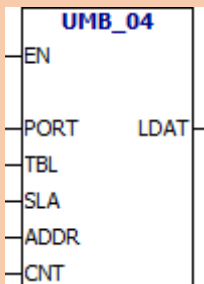
CNT: Número de registro de retentivo

LDAT: Armazena os dados que foram escritos a partir da estação escrava.

[Exemplo1](#)

[Exemplo2](#)

7.20.2. UMB-04



Lê registro de entrada.

EN: habilita ou desabilita

TBL: Tabela de configuração, se a entrada for MB200:
MB200 é a word de configuração:

(Instrução de saída) M200.0 Comunicação enfileirada

(Instrução de saída) M200.1 Comunicação completa

(Instrução de saída) M200.2 Erro de comunicação

(Instrução de saída) MB201 Número de erro: 0 indica que não há erros

SLA:MODBUS Endereço de escravo

ADDR: Ponto de ajuste de registrador retentivo (Ajuste de 3x)

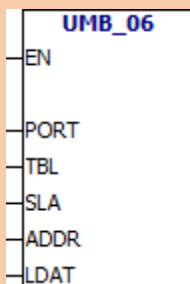
CNT: Número de registro de retentivo

LDAT: Armazena os dados que foram escritos a partir da estação escrava

[Exemplo1](#)

[Exemplo2](#)

7.20.3. UMB-06



Grava um único registrador retentivo

EN: habilita ou desabilita

TBL: Tabela de configuração, se a entrada for MB200:

MB200 é a word de configuração

(Instrução de saída) M200.0 Comunicação enfileirada

(Instrução de saída) M200.1 Comunicação completa

(Instrução de saída) M200.2 Erro de comunicação

(Instrução de saída) MB201 Número de erro: 0 indica que não há erros

SLA: MODBUS Endereço de escravo

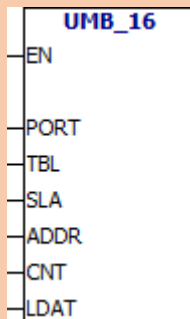
ADDR: Ponto de ajuste de registrador retentivo (Ajuste de 4x)

LDAT: Armazena os dados que foram escritos a partir da estação escrava

[Exemplo1](#)

[Exemplo2](#)

7.20.4. UMB-16



Grava mais de um registro retentivo.

EN: habilita ou desabilita

TBL: Tabela de configuração, se a entrada for MB200:

MB200 é a word de configuração:

(Instrução de saída) M200.0 Comunicação enfileirada

(Instrução de saída) M200.1 Comunicação completa

(Instrução de saída) M200.2 Erro de comunicação

(Instrução de saída) MB201 Número de erro: 0 indica que não há erros

SLA: MODBUS Endereço de escravo

ADDR: Ponto de ajuste de registrador retentivo (Ajuste de 4x)

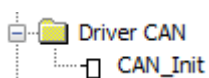
CNT: Número de registro de retentivo

LDAT: Armazena os dados que foram escritos a partir da estação escrava

[Exemplo1](#)

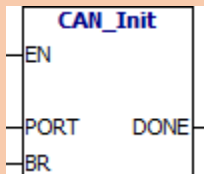
[Exemplo2](#)

7.21. CAN-Driver



[Como chamar o arquivo de ajuda](#)

7.21.1. CAN-Init



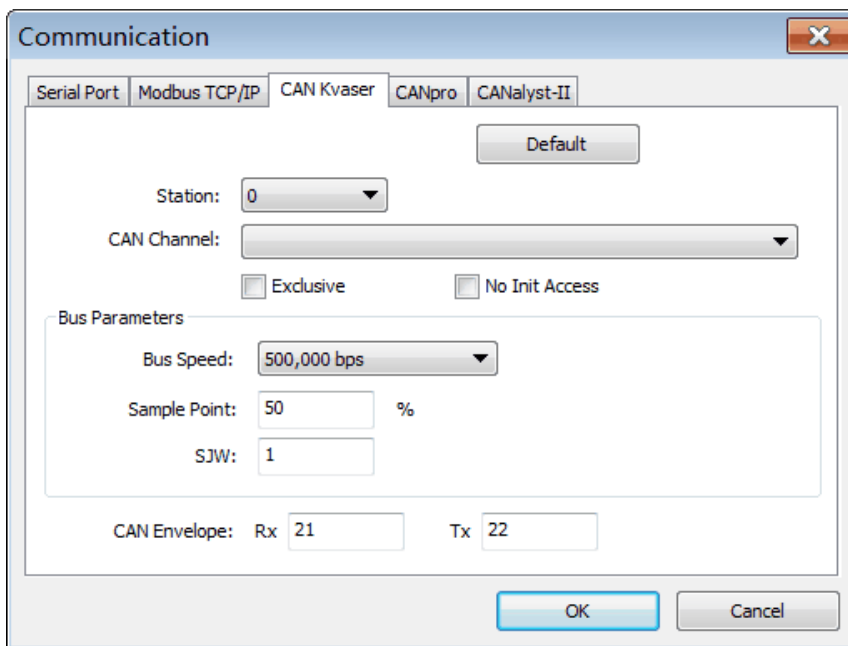
CAN_Init A instrução é usada para inicializar a porta da CAN.

EN: Se o valor de entrada for 1, a instrução inicializará a porta CAN.

PORT: número da porta, 0~1.

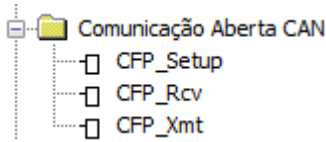
BR: Taxa de transmissão da porta CAN.

Você também pode definir esses parâmetros no software de programação.



Obs.: A comunicação CAN não é estável.
Use a comunicação serial.

7.22. Free-Port (CAN)



[Como chamar o arquivo de ajuda](#)

7.22.1. CFP-Setup

A comunicação CAN não é estável.
Use a comunicação serial.

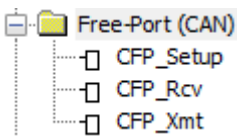
7.22.2. CFP-Rcv

A comunicação CAN não é estável.
Use a comunicação serial.

7.22.3. CFP-Xmt

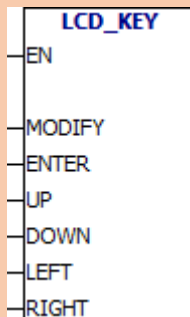
A comunicação CAN não é estável.
Use a comunicação serial.

7.23. LCD



[Como chamar o arquivo de ajuda](#)

7.23.1. LCD-KEY



LCD_KEY vincula LCD e variáveis chaves do CLP.

EN: Habilita

MODIFY: Modifica variáveis correspondentes.

ENTER: Confirma variáveis correspondentes

UP: Variável correspondente ao botão UP.

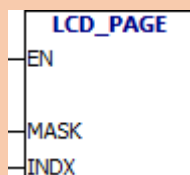
DOWN: Variável correspondente ao botão DOWN.

LEFT: Variável correspondente ao botão LEFT.

RIGHT: Variável correspondente ao botão RIGHT.

[Exemplo](#)

7.23.2. LCD-PAGE



LCD_PAGE vincula as instruções do LCD exibidas no display.

EN: Habilita

MASK: A máscara atual do grupo de páginas, é 1.

INDX: Número da página atual. Você pode modificar o número da página, o LCD exibirá a página.

Complementar: A entrada MASK é um byte.

Tome VB0 como um Exemplo:

VB0



Quando o bit 0 é igual a 1, o LCD exibirá 0 grupo.

Quando o bit 1 é igual a 1, o LCD exibirá 1 grupo.

·
·
·

Quando o bit 7 é igual a 1, o LCD exibirá 7 grupo.

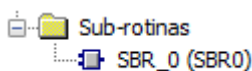
[Exemplo](#)

7.23.3. LCD-EDIT

<p>EN UNIT AUTO DEPTH LOOP FLASH V VMIN VMAX</p>	<p>LCD_EDIT: vincula as variáveis do CLP para editar o estado do LCD.</p> <p>EN: Habilita</p> <p>UNIT: Edita o número dos objetos na página.</p> <p>AUTO: Usa as teclas de LCD para editar.</p> <p>DEPTH: A edição profunda no objeto de edição.</p> <p>LOOP: editar LOOP.</p> <p>FLASH: O objeto de edição está piscando ou não.</p> <p>V: O valor atual do objeto de edição.</p> <p>VMIN: O valor mínimo do objeto de edição.</p> <p>VMAX: O valor máximo do objeto de edição.</p>
--	--

[Exemplo](#)

7.24. Sub-rotinas



[Como chamar o arquivo de ajuda](#)

7.24.1. Usando a sub-rotina

A sub-rotina é usada para o particionamento do programa. Quando o programa principal chama a sub-rotina e executa a sub-rotina, a sub-rotina executa todas as instruções para o fim. Então, o sistema retorna ao programa principal.

A sub-rotina é usada para o particionamento do programa. Ajuda a ler e gerenciar programas. Também ajuda a depurar e manter programas. Você pode usar o PLC de forma mais efetiva usando a sub-rotina. Por causa de todos os blocos de sub-rotinas não serem digitalizados quando não são chamados.

Se a sub-rotina apenas faz referência aos parâmetros e à memória local, então a sub-rotina pode ser movida. Para mover a sub-rotina, você não pode usar variáveis / símbolos globais (I, Q, M, SM, AI, AQ, V, T, C, S, endereço absoluto AC). Se a sub-rotina não chama parâmetros (IN, OUT ou IN_OUT) ou usa apenas variáveis locais, você pode exportar a sub-rotina e importá-la para outro projeto.

Condições de utilização da sub-rotina:

1. Crie uma sub-rotina
 2. Defina parâmetros na tabela de variáveis local.
 3. Chame a sub-rotina da POU apropriada (do programa principal ou outra sub-rotina)
- Usar a sub-rotina não guarda ou restaura o acumulador.

7.24.2. Usando parâmetros para chamar sub-rotina

A sub-rotina pode conter os parâmetros de transferência. O parâmetro é definido na tabela de variáveis locais da sub-rotina.

Os parâmetros devem possuir um nome de símbolo (até 23 caracteres), um tipo de variável e um tipo de dado. Cada sub-rotina pode ser configurada em até 16 parâmetros IN / OUT.

A tabela de variáveis locais possui 4 tipos de variáveis. São IN, IN-OUT, OUT e TEMP.

		Symbol	Var Type	Data Ty...	Comment
	✓ L0.0	ss	IN	BOOL	
			IN	BOOL	
			IN_OUT	BOOL	
			IN_OUT	BOOL	
			IN_OUT	BOOL	
			OUT	BOOL	
			TEMP	BOOL	

Tipo e descrição do parâmetro

IN Os parâmetros são transferidos para a sub-rotina. Se o parâmetro for um endereço direto (por exemplo, VB10), o valor da localização especificada é transferido para a sub-rotina. Se o parâmetro for um Endereço Indireto (como * AC1), o valor da localização especificada é transferido para a sub-rotina. Se o parâmetro for a constante de dados (16 # 1234) ou endereço (& VB100), constantes ou endereços são transferidos para a sub-rotina.

IN_OUT O valor da localização especificada é transferido para a sub-rotina. O resultado da operação da sub-rotina é transferido para a mesma localização especificada. Este parâmetro não permite o uso de constantes (como 16 # 1234) e endereços (por exemplo, e VB100).

OUT O resultado da operação da sub-rotina é transferido para o local especificado. Os contatos (como 16 # 1234) e os endereços (por exemplo, e VB100) não podem ser usados como saída.

TEMP Qualquer memória local que não seja usada como um parâmetro de transferência não pode ser usada para armazenamento temporário na sub-rotina.

<u>Booleana</u>	É usado para entrada e saída.
<u>Byte, word, double word</u>	Parâmetros de entrada ou saída sem sinal.
<u>Integer, double integer</u>	Parâmetros de entrada ou saída com sinal.
<u>Número Real</u>	Identifica valores de ponto flutuante de precisão.
<u>String</u>	Este Tipo de dado é usado como um ponteiro de quatro bytes para a string.
<u>Enable bit</u>	O bit de habilitação booleana pode ser usado apenas para bit. Pode ser usado como entrada.

7.24.3. Como configurar uma sub-rotina

Os seguintes métodos podem ser usados para estabelecer uma sub-rotina:

1. Gerenciador de projeto → bloco de programa → Botão direito do bloco de programa → Inserir → sub-rotina.
2. Gerenciador de projeto → bloco de programa → SBR-0 → Clique com o botão direito do mouse em SBR-0 → Inserir → sub-rotina.

Você pode usar a tabela de variáveis local para definir os parâmetros da sub-rotina.

Notas :

1. Lembre-se de que cada POU no programa possui uma tabela de variáveis local independente. Na sub-rotina **A**, você só pode usar a tabela de variáveis, local para definir variáveis.
2. Qualquer sub-rotina pode ser configurada para 16 parâmetros IN / OUT. Se o número de parâmetros for maior que 16, o programa gerará erros.
3. Você pode escrever uma sub-rotina na janela de edição do programa.

4. Clique no rótulo da POU que deseja editar. SOU você pode editar a POU na janela de edição do programa.

O editor insere as instruções de terminação da POU automaticamente. (END para MAIN, RET para SBR, RETI para INT).

7.24.4. Como chamar uma sub-rotina

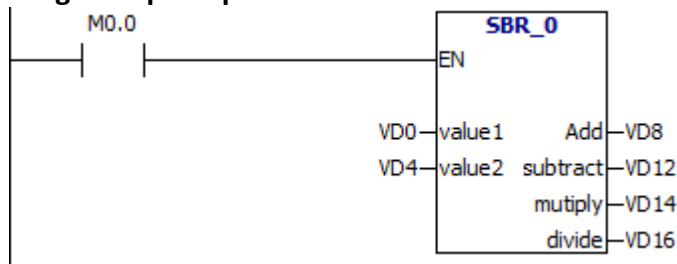
Você pode chamar uma sub-rotina no programa principal, outra sub-rotina ou uma rotina de interrupção; Você não pode chamar a sub-rotina da própria sub-rotina.

Passos para chamar a sub-rotina:

1. Na janela de edição do programa, coloque o cursor na posição em que deseja colocar a sub-rotina.
2. Instruções → Subrotina, depois selecione a sub-rotina que você precisa. Clique duas vezes nele.

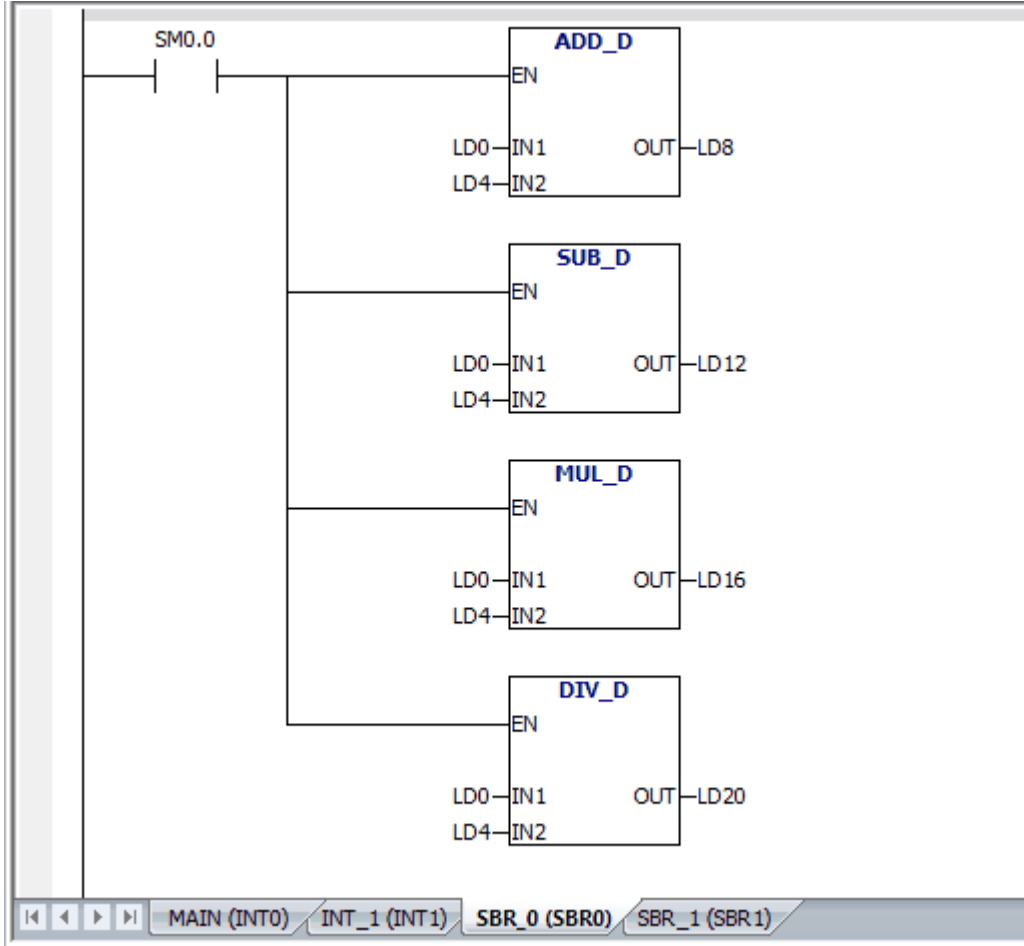
Exemplo : Quatro operações aritméticas

Programa principal:



Sub-rotina:

Program Editor				
	Symbol	Var Type	Data Type	Comment
✓ LD0	value1	IN	DINT	
✓ LD4	value2	IN	DINT	
		IN	BOOL	
		IN_OUT	BOOL	
✓ LD8	Add	OUT	DINT	
✓ LD12	subtract	OUT	DINT	
✓ LD16	multiply	OUT	DINT	
✓ LD20	divide	OUT	DINT	
		OUT	BOOL	
		TEMP	BOOL	



8. Área de armazenamento do CLP

8.1. Tipos e propriedades da área de armazenamento

Região	Descrição	Bit	Byte	Word	Double Word	Retentivo	Forçar
I	Entrada discreta e registro de imagens	Lê / Grava	Lê / Grava	Lê / Grava	Lê / Grava	Não	Sim
Q	Saída discreta e registro de imagens	Lê / Grava	Lê / Grava	Lê / Grava	Lê / Grava	Não	Sim
M	Bit de memória interna	Lê / Grava	Lê / Grava	Lê / Grava	Lê / Grava	Sim	Sim
SM	Bit de memória especial	Lê / Grava	Lê / Grava	Lê / Grava	Lê / Grava	Não	Não
V	Memória variável	Lê / Grava	Lê / Grava	Lê / Grava	Lê / Grava	Sim	Sim
T	Valor atual do temporizador e bit do temporizador	Lê / Grava (T bit)	Não	Lê / Grava (T Current value)	Não	Sim	T bit
C	Bit de contagem e valor atual do contador	Lê / Grava (C bit)	Não	Lê / Grava (C current value)	Não	Valor atual C Sim	Não
HC	Valor atual do contador de alta velocidade	Não	Não	Não	Read-only	Não	Não
AI	Entrada Analógica	Não	Não	Read-only	Não	Não	Sim
AQ	Saída Analógica	Não	Não	write only	Não	Não	Sim
AC	Registro de acumulador	Não	Lê / Grava	Lê / Grava	Lê / Grava	Não	Não
L	Memória variável local	Lê / Grava	Lê / Grava	Lê / Grava	Lê / Grava	Não	Não
S	SCR	Lê / Grava	Lê / Grava	Lê / Grava	Lê / Grava	Não	Não

8.2. Endereçamento direto e indireto

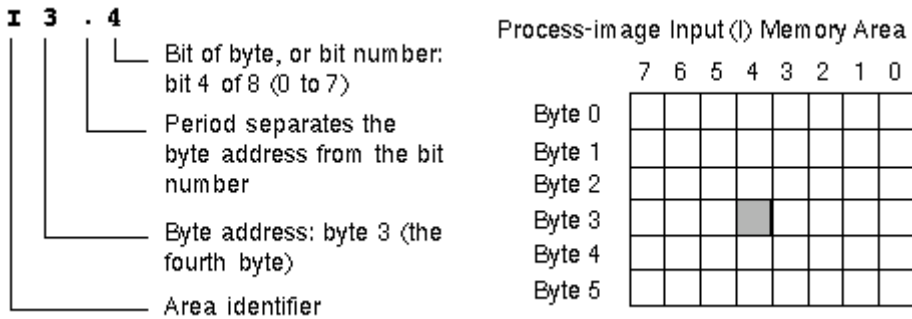
Quando você escreve o programa, você pode usar três maneiras de abordar as instruções:

1. Endereçamento direto
2. Redição de símbolos
3. Índice Indireto

Endereçamento direto

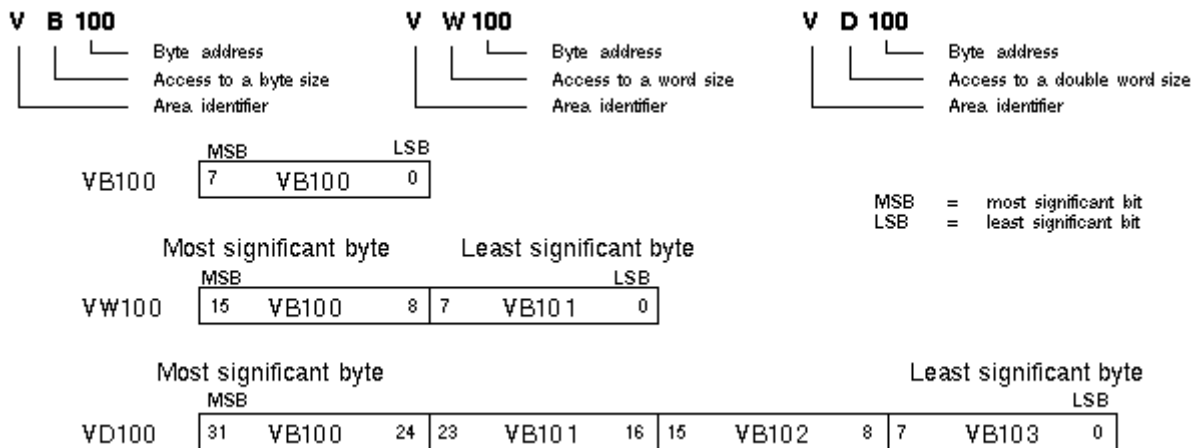
O CLP pode especificar diretamente a área, o tamanho e a localização da memória. Para ler / gravar um pouco na área da memória, é necessário especificar o endereço. O endereço inclui o identificador da área de memória, o endereço do byte, um período e um número.

Exemplo:



Especificar endereços de bytes, word e double word são semelhantes a especificar o endereço do bit.

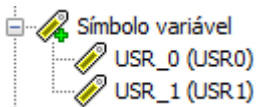
Exemplo:



Endereçamento simbólico

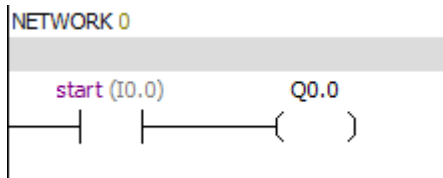
O endereçamento de símbolos consiste em letras, números e caracteres.

Você pode definir o símbolo do endereço pelas seguintes etapas:



	Symbol	Address	Data Ty...	Comment
✓	start	I0.0	BOOL	start
			BOOL	
			BOOL	
			BOOL	

Você pode inserir "start" como o endereço do ponto de entrada (I0.0).

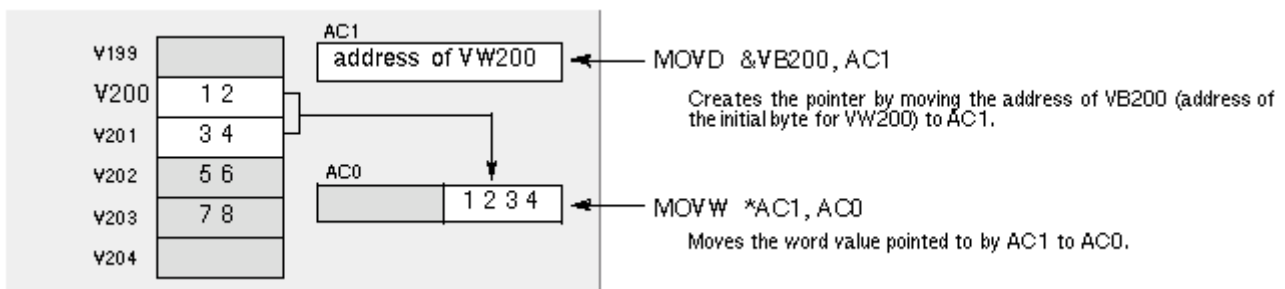


Endereço Indireto

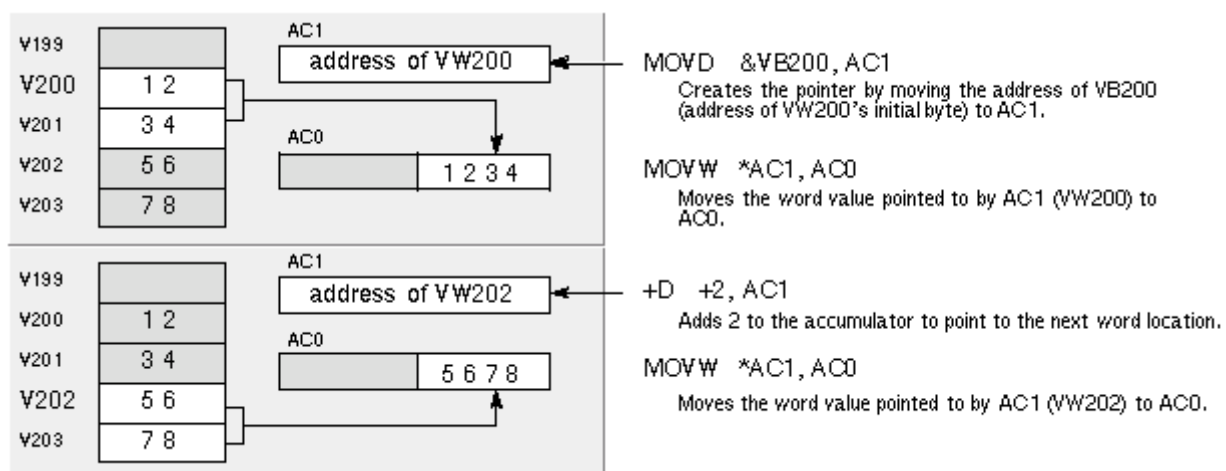
Endereço indireto usa ponteiro para acessar os dados da memória. O ponteiro é uma double word. Ele contém o endereço de uma outra localização de memória. Apenas a localização da memória V, L ou o acumulador de registro (AC1, AC2, AC3) podem ser usados como ponteiros. O CLP permite que o ponteiro acesse a seguinte área de memória: I, Q, V, M, S, T, CT e C. Só pode usar o valor atual. Ponteiro consiste na localização da memória e no símbolo "&". Para especificar o operando ser um ponteiro, você deve inserir um asterisco (*) na frente do operando.

Exemplo:

Os valores armazenados no VB200 e VB201 são movidos para AC0.



Conforme mostrado na figura abaixo, você pode alterar o valor do ponteiro. Como o ponteiro é um valor de 32 bits, você deve usar as instruções da double word para modificar o valor do ponteiro.



Pronto:

Se você usar o ponteiro para executar a operação do byte, o intervalo mínimo do ponteiro é 1.

Se você usar o ponteiro para executar a operação da word, o intervalo mínimo do ponteiro é 2.

Se você usar o ponteiro para executar a operação da double word, o intervalo mínimo do ponteiro é 4.

Se o valor do ponteiro for maior do que o valor máximo da memória V, o programa irá gerar erros.

O valor atual do temporizador e do contador é de 16 bits, de modo que o intervalo mínimo do ponteiro é 2.

8.3. Bit, byte, word e double word

Bit access

Se você quiser acessar um bit, você precisa especificar o endereço do bit. O endereço contém o identificador de região e o número do byte. Zero é o primeiro endereço de todas as áreas de dados. O ponto decimal é usado para separar o número de bytes e o número de bits. O intervalo do número de bits é 0 ~ 7. Para Exemplo: M0.0 ~ M0.7

Byte, word e double word

Se você quiser acessar byte, word ou double word, você precisa especificar o endereço. A endereço contém um identificador de região, uma letra e um número de endereço.

Exemplo:

VB100 Acesso de memória endereço byte V 100
 VW100 Acesso de memória endereço byte V 100 e 101
 VD100 Acesso de memória endereço byte V 100, 101, 102 e 103

8.4. Alcance do endereço da memória

NEX18

	Bit	Byte	Word	Double Word
I	I0.0~I31.7	IB IB0~IB31	IW IW0~IW30	ID ID0~ID28
Q	Q0.0~Q31.7	QB QB0~QB31	QW QW0~QW30	QD
M	M0.0~31.7	MB0~MB31	MW MW0~MW30	MD MD0~MD28

S		SB	SB0~SB31	SW	SW0~SW30	SD	SD0~SD28
SM	SM0.0~SM551.7	SMB	SMB0~SMB551	SMW	SMW0~SMW550	SMD	SMD0~SMD548
T	T0~T255			T	T0~T255		
C	C0~C255				C0~C255		
V	V0.0~V8191.7	VB	VB0~VB8191	VW	VW0~VW8190	VD	VD0~VD8188
L	L0.0~L63.7	LB	LB0~LB63	LW	LW0~LW62	LD	LD0~LD60
		AC	AC0~AC3	AC	AC0~AC3	AC	AC0~AC3
						HC	HC0~HC15
				AIW	AIW0~AIW180		
				AQW	AQW0~AQW180		

NEX14

	Bit		Byte		Word		Double Word
I	I0.0~I31.7	IB	IB0~IB31	IW	IW0~IW30	ID	ID0~ID28
Q	Q0.0~Q31.7	QB		QW	QW0~QW30	QD	QD0~QD28
M	M0.0~31.7		MB0~MB31	MW	MW0~MW30	MD	MD0~MD28
S	S0.0~S31.7	SB	SB0~SB31	SW	SW0~SW30	SD	SD0~SD28
SM	SM0.0~SM551.7	SMB	SMB0~SMB551	SMW	SMW0~SMW550	SMD	SMD0~SMD548
T	T0~T255			T	T0~T255		
C	C0~C255			C	C0~C255		
V	V0.0~V4095.7	VB	VB0~VB4095		VW0~VW4094	VD	VD0~VD4092
L	L0.0~L63.7	LB	LB0~LB63	LW	LW0~LW62	LD	LD0~LD60
		AC	AC0~AC3	AC	AC0~AC3	AC	AC0~AC3
						HC	HC0~HC15
				AIW	AIW0~AIW180		
				AQW	AQW0~AQW180		

8.5. Tipo de dados

Tipo de dado	Tamanho do Dado	Alcance
BOOL		0~1
BYTE	8	16#00~16#FF
WORD	16	
DWORD	32	16#00000000~16#FFFFFFFF
SINT	8	-128~127
	16	-32768~32767
DINT	32	-2147483648~2147483647
USINT	8	0~255
UINT	16	0~65535
	32	0~4294967295

8.6. Constante

Tamanho do Dado:	Intervalo Unsigned Integer		Intervalo Signed integer	
	Digito Decimal	Digito Hexadecimal	Digito Decimal	Digito Hexadecimal
B (byte)	0~255		-128 ~+127	80~7F
W (word)	0~65535	0~FFFF	-32768~+32767	8000~7FFF
D (double word)	0~4294967295	0~FFFF FFFF	-2147483648~+2147483647	8000 0000~7FFF FFFF

Tamanho do Dado:	Número Decimal (+)	Número Decimal (-)
D (double word)	1.175495E-38 a 3.402823E+38	(-)1.175495E-38 à (-)3.402823E+38

9. Atribuição e função da área de armazenamento especial SM

9.1. Área de armazenamento especial SM

SMB0

Always_On	SM0.0	Sempre ON
First_Scan_On	SM0.1	ON apenas para o primeiro ciclo de digitalização
Clock_60s	SM0.4	30 segundos OFF, 30 segundos ON
Clock_1s	SM0.5	0,5 segundo OFF, 0,5 segundo ON

SMB1

Result_0	SM1.0	Define 1 para execução de instruções quando o resultado da operação = 0
Overflow_Illegal	SM1.1	Define 1 para execução de instruções sobre o estouro ou valor inválido.
Neg_Result	SM1.2	Define 1 quando uma operação matemática produz um Resultado Negativo
Divide_By_0	SM1.3	Define 1 quando uma tentativa é feita para dividir por zero
Table_Overflow	SM1.4	Define 1 quando a instrução Add à tabela tenta sobrecarregar a tabela
Table_Empty	SM1.5	Define 1 quando uma instrução LIFO ou FIFO tenta ler a partir de uma tabela vazia
Nãot_BCD	SM1.6	Define 1 quando uma tentativa é feita para converter um valor Non-BCD para um valor binário
Nãot_Hex	SM1.7	Define 1 quando um valor ASCII não pode ser convertido em um valor hexadecimal válido

Os endereços das variáveis das teclas do LCD do CLP:

F1	→	SM191.0
F2	→	SM191.1
F3	→	SM191.2
F4	→	SM191.3
ESC	→	SM190.0
OK	→	SM190.1
UP	→	SM190.2
DOWN	→	SM190.3
LEFT	→	SM190.4
RIGHT	→	SM190.5

Quando o valor de SM192.0 for igual a 1, o LCD ficará claro.

Quando o valor de SM192.0 é igual a 0, o LCD ficará escuro.

Tempo de Scan SMW22-SMW26

SMW22	Tempo do último scan
SMW24	Tempo mínimo de scan
SMW26	Tempo máximo de scan

10. Comunicação Nexo Ladder

10.1. CLP NEXO - Introdução básica de comunicação de rede

O CLP Nexo é projetado para resolver suas necessidades de comunicação e rede. Ele suporta redes simples e redes complexas. Ladder amigável, torna simples configurar e configurar sua rede.

Definição de rede Master ou Slave

O CLP Nexo suporta a rede mestre escravo. Pode ser usado como a estação mestre ou a estação escrava na rede. O Ladder é sempre usada como a estação mestre.

Estação mestre: a estação mestre pode enviar uma solicitação para um outro dispositivo na rede. A estação principal também pode responder aos pedidos de outras estações mestres na rede.

Estação escrava: o dispositivo configurado para ser a estação escrava só pode responder aos pedidos de uma estação mestre; a estação escrava não tomará a iniciativa de emitir uma solicitação.


O conceito de taxa de transmissão e endereço de rede

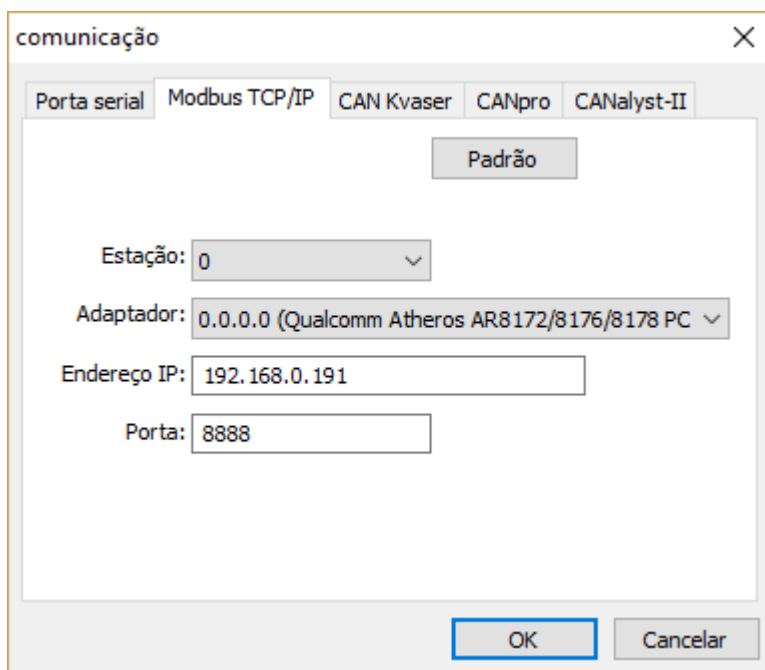
A taxa de transmissão de dados na rede é chamada de baud rate. São em kbaud e Mbaud. Por Exemplo, 19.2 kbaud, indica que 19200 bits são transmitidos por segundo.

Cada dispositivo deve ter a mesma taxa de transmissão na rede. Assim, a taxa de comunicação da rede é decidida pela taxa de transmissão mínima do equipamento.

O intervalo da taxa de transmissão do CLP é 1200 bps ~ 115200 bps. O valor padrão é 9600bps.

Defina a taxa de transmissão e o endereço da rede

Abra a comunicação no gerenciamento de projetos:  comunicação



Você pode definir o número da estação, a porta, a taxa de transmissão, a paridade e o bit de parada. O número da estação padrão é 0. A taxa de transmissão padrão é de 9600 bps.

Definir a taxa de transmissão e o endereço de rede do CLP

Bloqueio do sistema aberto no gerenciamento de projetos  Bloco de sistema

Bloco do Sistema

Memória Retentiva	Interrupção	Forçar Saídas
RS232/RS485	RS232/RS485	CAN Senha

Padrão

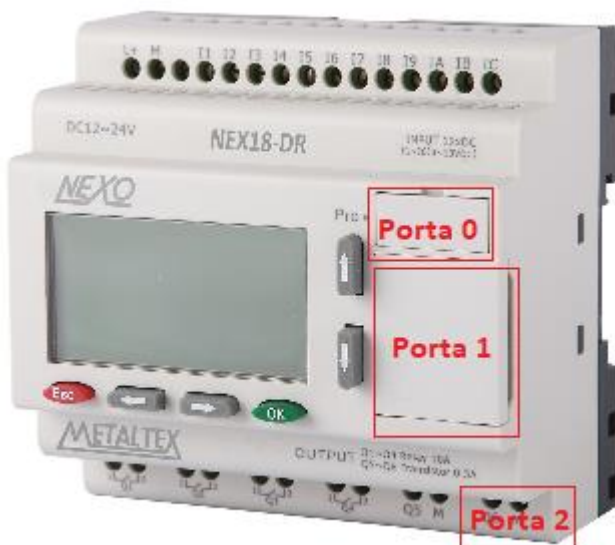
	Porta 0	Porta 1
Protocolo:	Modbus	Modbus
Estação:	1	1 (Intervalo 0... 255)
Velocidade:	9600 bps	9600 bps
Bit dados:	8 (RTU)	8 (RTU)
Paridade:	NONE	NONE
b.Parada:	1 Bit	1 Bit
Tempo Resposta (100ms)	10	10 (Intervalo 1... 255)
Intervalo de quadros (B)	10	10 (Intervalo 1... 255)

As modificações só terão efeito após o Download

OK Cancelar

Você pode definir o número da estação, taxa de transmissão, bits de dados, bit de paridade e bit de parada do PLC.

10.2. Porta de Comunicação



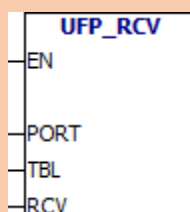
10.3. Comunicação do CLP

O CLP Nexo suporta comunicação de porta livre, comunicação MODBUS e comunicação CAN.

10.3.1. Comunicação de porta livre

A comunicação de porta livre é uma comunicação half duplex baseada na comunicação RS-485. Os usuários podem fazer seu próprio protocolo de comunicação na comunicação de porta livre. Os dispositivos de terceiros usam principalmente a comunicação serial RS-485. O núcleo da comunicação de porta livre está recebendo e enviando instruções. A comunicação RS-485 não pode receber e enviar dados ao mesmo tempo. O formato de comunicação RS-485 inclui um bit de início, caracteres de 7 ou 8 bits, um bit de paridade e um bit de parada.

Instrução de porta livre



UFP_RCV: Recebe instrução de dados.

PORT: Porta de comunicação.

TBL: Tabela de configuração, se a entrada for MB200.

MB200 é o byte de configuração:

(Saída de instrução) M200.0 Comunicação pronta.

(Saída de instrução) M200.1 Comunicação completa

(Saída de instrução) M200.2 Erro de comunicação

(Entrada de instrução) M200.3 Confirma envio de CRC

(Entrada de instrução) M200.4 Confirma envio de CRC

(Entrada de instrução) M200.5 Confirma recebimento de CRC

(Entrada de instrução) M200.6 Confirma recebimento de CRC

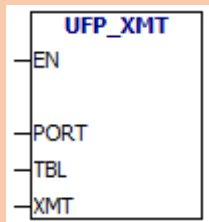
(Saída de instrução) MB201 Número de erro : 0 indica que não há erros.

RCV: recebe dados, se a entrada for MB400:

(Entrada de instrução) MW400: Tamanho de buffer FIFO de recebimento de dados (unidade de byte)

(Saída de instrução) MW402: Tamanho dos dados recebidos (em bytes)

(Saída de instrução) MB404 ~ ... recebe dados.



UFP_XMT: Recebe instrução de dados.

PORT: Porta de comunicação.

TBL: Tabela de configuração, se a entrada for MB200.

MB200 é o byte de configuração:

(Saída de instrução) M200.0 Comunicação pronta.

(Saída de instrução) M200.1 Comunicação completa

(Saída de instrução) M200.2 Erro de comunicação

(Entrada de instrução) M200.3 Confirma envio de CRC

(Entrada de instrução) M200.4 Confirma envio de CRC

(Entrada de instrução) M200.5 Confirma recebimento de CRC

(Entrada de instrução) M200.6 Confirma recebimento de CRC

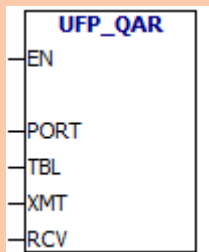
(Saída de instrução) MB201 Número de erro : 0 indica que não há erros.

XMT: envia dados FIFO, se a entrada for MB400:

(Entrada de instrução) MW400 envia dados FIFO tamanho do buffer (unidade de byte)

(Entrada de instrução) MW402 Enviando tamanho de dados (byte)

(Entrada de instrução) MB404 ~ ... Enviar dados.



UFP_QAR: Recebe e envia instrução de dados.

PORT: Porta de comunicação.

TBL: Tabela de configuração, se a entrada for MB200.

MB200 é o byte de configuração:

(Saída de instrução) M200.0 Comunicação pronta.

(Saída de instrução) M200.1 Comunicação completa.

(Saída de instrução) M200.2 Erro de comunicação.

(Entrada de instrução) M200.3 Confirma envio de CRC

(Entrada de instrução) M200.4 Confirma envio de CRC

(Entrada de instrução) M200.5 Confirma recebimento de CRC

(Entrada de instrução) M200.6 Confirma recebimento de CRC

(Saída de instrução) MB201 Número de erro : 0 indica que não há erros.

XMT: Envia dados FIFO, se a entrada for MB300:

(Entrada de instrução) MW300 Sending data FIFO buffer size (byte unit)

(Entrada de instrução) MW302 Sending data size (byte)

(Entrada de instrução) MB304 ~ ... Send data.

RCV: Recebe dados FIFO, se a entrada for MB400:

(Entrada de instrução) MW400 recebe dados FIFO tamanho do buffer (unidade de byte)

(Saída de instrução) MW402 tamanho de dados recebidos (em byte)

(Saída de instrução) MB404 ~ ... Dados recebidos.

UFP_RCV、UFP_XMT、UFP_QAR número dos erros:

- ✂1 Porta inexistente
- ✂2 A porta não está habilitada
- ✂3 Fila de tarefa de comunicação esta cheia
- ✂4 Erro de tabela
- ✂5 Erro no envio de dados
- ✂6 Tempo limite
- ✂7 Erro no recebimento de dados
- ✂8 Erro de verificação de dados de recepção

10.3.2. Comunicação MODBUS

O protocolo MODBUS é comumente usado nos controladores. Os diferentes dispositivos podem se comunicar usando o protocolo de comunicação MODBUS. Tornou-se um padrão industrial em geral. Você pode usá-lo para conectar diferentes dispositivos.

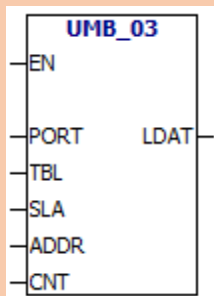
Este protocolo define uma estrutura de mensagens, não importa qual rede eles usam para se comunicar. Ele descreve o processo do controlador que solicita acessar outros dispositivos. Formulou a estrutura do domínio da mensagem e o formato comum do conteúdo.

O protocolo de rede MODBUS determina que cada controlador deve deixar seu endereço. Ele identifica as mensagens enviadas de diferentes endereços e decide quais ações devem ser tomadas. O controlador gera informações de feedback, o formato da informação é o formato de informação de MODBUS. Ele é emitido através do protocolo MODBUS.

O endereço MODBUS geralmente contém tipos de dado e offset. O endereço MODBUS contém um total de 5 caracteres. O primeiro caractere representa o Tipo de dado e os outros quatro caracteres representam os valores corretos no Tipo de dado.

Instruções MODBUS:

UMB03



UMB_03: Lê mais que um registro retentivo.

EN: Habilita ou desabilita.

TBL: Tabela de configuração, se a entrada for MB200:

MB200 é a word de configuração.

(Saída de instrução) M200.0 A comunicação foi colocada em fila.

(Saída de instrução) M200.1 Comunicação pronta.

(Saída de instrução) M200.2 Erro de comunicação.

(Saída de instrução) MB201 Número de erro : 0 indica que não há erros.

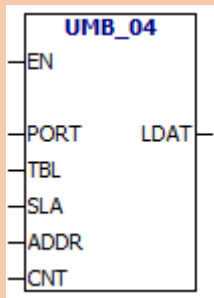
SLA: endereço MODBUS slave.

ADDR: o offset de registro retentivo (Offset de 4x)

CNT: Número de registro retentivo.

LDAT: Armazena os dados que foram escritos a partir da estação slave.

UMB04



UMB_04: Lê registro de entrada

EN: habilita ou desabilita.

TBL: Tabela de configuração, se a entrada for MB200: MB200 é a word de configuração.

(Saída de instrução) M200.0 A comunicação foi colocada em fila.

(Saída de instrução) M200.1 Comunicação pronta.

(Saída de instrução) M200.2 Erro de comunicação.

(Saída de instrução) MB201 Número de erro : 0 indica que não há erros.

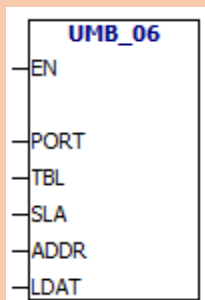
SLA: endereço MODBUS slave.

ADDR: o offset de registro retentivo (Offset de 3x)

CNT: Número de registro retentivo.

Armazena os dados que foram escritos a partir da estação slave.

UMB06



UMB_06: Grava um registro retentivo.

EN: habilitado ou desabilitado.

TBL: Tabela de configuração, se a entrada for MB200: MB200 é a word de configuração.

(Saída de instrução) M200.0 A comunicação foi colocada em fila.

(Saída de instrução) M200.1 Comunicação pronta.

(Saída de instrução) M200.2 Erro de comunicação.

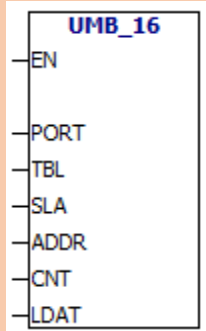
(Saída de instrução) MB201 Número de erro : 0 indica que não há erros.

SLA: endereço MODBUS slave.

ADDR: o offset de registro retentivo (Offset de 4x)

LDAT: Armazena os dados que foram escritos a partir da estação slave.

UMB16



UMB_16: Grava mais de um registro retentivo.

EN: habilitado ou desabilitado.

TBL: Tabela de configuração, se a entrada for MB200: MB200 é a word de configuração.

(Saída de instrução) M200.0 A comunicação foi colocada em fila.

(Saída de instrução) M200.1 Comunicação pronta.

(Saída de instrução) M200.2 Erro de comunicação.

(Saída de instrução) MB201 Número de erro : 0 indica que não há erros.

SLA: endereço MODBUS slave.

ADDR: o offset de registro retentivo (Offset de 4x)

CNT: Número de registro retentivo.

LDAT: Armazena os dados que foram escritos a partir da estação slave.

ENDEREÇO MODBUS

ENDEREÇO	I0.0~I31.0	Q0.0~Q31.0	AIW0~AIW17 8	VW0~VW819 0
ENDEREÇO MODBUS	0 ~ 255	0 ~ 255	0 ~ 89	0 ~ 4095

Exemplo:

VW0 MODBUS endereço 0

VW2 MODBUS endereço 1

VW4 MODBUS endereço 2

.

.

.

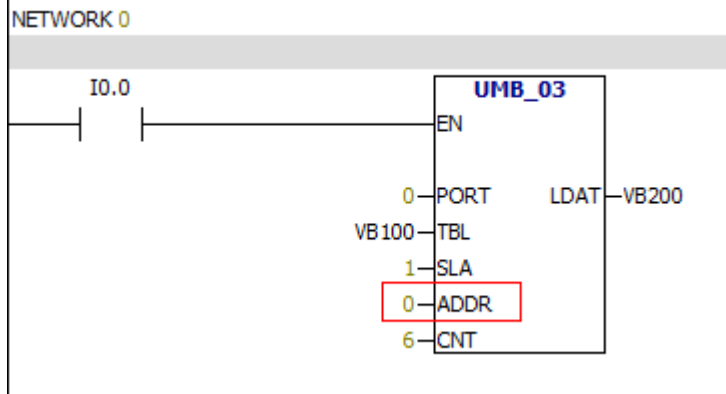
VW4094 MODBUS endereço 2047

VW8190 MODBUS endereço 4095

Atenção:

AIW3, AIW5..... e VW3, VW5, VW7 não possuem endereço MODBUS.

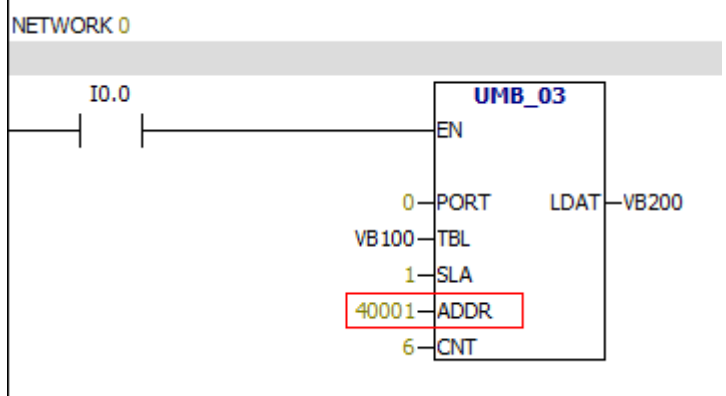
Exemplo:



0→VW0

1→VW2

Segue abaixo o exemplo do erro:



O endereço 40001 ≠ vw0

[Exemplo](#)

10.3.3. Comunicação CAN

A comunicação da CAN não é mais estável. Escolha outros meios de comunicação.

10.3.4. Desempenho de rede

Os seguintes fatores afetarão o desempenho da rede (taxa de transmissão e a estação mestre produzem o maior impacto para o desempenho da rede):

Baud rate: Ele determina a velocidade da comunicação de rede.

Número de estações mestre na rede: Para melhorar o desempenho da rede, você pode reduzir o número de estações principais na rede. Cada estação na rede aumentará os requisitos adicionais da rede.

Selecione a estação mestre e os endereços da estação escrava: O endereço da estação principal deve ser contínuo. Quando houver um endereço de espaçamento entre as estações mestres, a estação mestre verificará o endereço de espaçamento incessantemente e não verá se há uma estação principal esperando que esteja na linha. Assim, o endereço de espaçamento da estação principal aumentará Os requisitos adicionais da rede. Você pode definir o endereço do escravo para qualquer valor. Mas o endereço da estação escrava não pode ser colocado entre os endereços da estação principal. Ou aumentará os requisitos adicionais da rede.

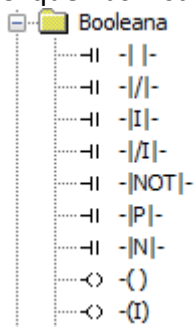
11. Capítulo Adicional

11.1. Como chamar o arquivo de ajuda

Para chamar o arquivo de ajuda, você deve clicar nas instruções e, em seguida, pressionar F1.

Exemplo:

Clique nas instruções NA.



Em seguida, pressione F1, o arquivo de ajuda será chamado.

Normalmente Aberta e normalmente fechada



Input/output	Operand	Data type
Bit (LAD、STL)	I, Q, M, SM, T, C, V, S, L	Boolean
Input (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Boolean
Output (FBD)	I, Q, M, SM, T, C, V, S, L, Enable bit	Boolean

<p>Normally open and normally closed</p>	<p>When the bit is equal to 1, the normally open contact is closed, and the normally closed contact is disconnected.</p> <p>When the bit is equal to 0, the normally open contact is disconnected, and the normally closed contact is closed.</p> <p>In STL, the normally open contact is represented by "LD", "And" and "Or" instructions.</p> <p>In STL, normally closed contacts are represented by "NOT", "NOT AND" and "NOT OR" instructions.</p>
--	--

11.2. Como forçar os pontos de entrada e saída

Como forçar pontos de entrada

1. Monitora seu programa
2. Na tabela de status, mude de "não forçado" para "forçado"

	Address	Data Ty...	Value	Forced
	I0.0	BOOL	0	Unforced
	I0.1	BOOL	0	Unforced
	I0.2	BOOL	0	Unforced
	I0.3	BOOL	0	Unforced
	I0.4	BOOL	0	Unforced



Status Chart

	Address	Data Ty...	Value	Forced
	I0.0	BOOL	0	Forced
	I0.1	BOOL	0	Unforced
	I0.2	BOOL	0	Unforced
	I0.3	BOOL	0	Unforced

Em seguida, digite 1 para o "valor"

Status Chart

	Address	Data Ty...	Value	Forced
	I0.0	BOOL	1	Forced
	I0.1	BOOL	0	Unforced
	I0.2	BOOL	0	Unforced
	I0.3	BOOL	0	Unforced
	I0.4	BOOL	0	Unforced

O ponto de entrada será forçado a 1.

Se você inserir 0 para o "valor", o ponto de entrada será forçado a 0.

Como forçar os pontos de saída

1. Monitora seu programa
2. Na tabela de status, mude de "não forçado" para "forçado"

Status Chart

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
Q0.0	BOOL	0	Unforced				
Q0.1	BOOL	0	Unforced				
Q0.2	BOOL	0	Unforced				
Q0.3	BOOL	0	Unforced				
Q0.4	BOOL	0	Unforced				
Q0.5	BOOL	0	Unforced				
Q0.6	BOOL	0	Unforced				
Q0.7	BOOL	0	Unforced				



Status Chart

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
Q0.0	BOOL	0	Forced				
Q0.1	BOOL	0	Unforced				
Q0.2	BOOL	0	Unforced				
Q0.3	BOOL	0	Unforced				
Q0.4	BOOL	0	Unforced				
Q0.5	BOOL	0	Unforced				
Q0.6	BOOL	0	Unforced				
Q0.7	BOOL	0	Unforced				

3. Na tabela de status, a entrada 1 para o "valor"

Status Chart

Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced
Q0.0	BOOL	1	Forced				
Q0.1	BOOL	0	Unforced				
Q0.2	BOOL	0	Unforced				
Q0.3	BOOL	0	Unforced				
Q0.4	BOOL	0	Unforced				
Q0.5	BOOL	0	Unforced				
Q0.6	BOOL	0	Unforced				
Q0.7	BOOL	0	Unforced				

O ponto de saída será forçado a 1.

Se você inserir 0 para o "valor", o ponto de saída será forçado a 0.

11.3. Faixa de valor da analógica

0 à 10v → 0 ~ 1000

0 à 20ma → 0 ~ 1000

4 à 20ma → 0 ~ 1000

-50 à 200°C → -500 ~ 2000

Quando você usa o módulo PT100, o intervalo de valores da quantidade analógica é -500 ~ 2000. Ele corresponde à temperatura que é -50 à 200 °C.

11.4. Endereço do módulo de expansão

Você pode usar o interruptor de discagem para definir o endereço. O endereço de cada módulo de extensão não pode ser o mesmo.

Tabela de endereço de entrada digital:

Endereço da Expansão	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Início do endereço	I2.0	I3.0	I4.0	I5.0	I6.0	I7.0	I8.0	I9.0	I10.0	I11.0	I12.0	I13.0	I14.0	I15.0	I16.0	I17.0

Tabela de endereço de saída digital:

Endereço da Expansão	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Início do endereço	Q2.0	Q3.0	Q4.0	Q5.0	Q6.0	Q7.0	Q8.0	Q9.0	Q10.0	Q11.0	Q12.0	Q13.0	Q14.0	Q15.0	Q16.0	Q17.0

Tabela de endereço de saída analógica:

Endereço da Expansão	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Início do endereço	AQW2.0	AQW3.0	AQW4.0	AQW5.0	AQW6.0	AQW7.0	AQW8.0	AQW9.0	AQW10.0	AQW11.0	AQW12.0	AQW1.30	AQW1.40	AQW1.50	AQW1.60	AQW170

Tabela de endereço de entrada analógica:

Endereço da Expansão	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Início do endereço	AIW 20	AI W3 0	AI W4 0	AI W5 0	AI W6 0	AIW 70	AIW 80	AI W9 0	AIW 100	AIW 110	AIW 120	AIW 130	AIW 140	AIW 150	AIW 160	AIW 170

11.5. Formulário de endereço do módulo de expansão

Formulário de endereço do módulo de expansão digital:

Início do Endereço da Entrada digital = I (Extension Address+1) .0

Início do Endereço da Saída digital = Q (Extension Address+1) .0

Formulário de endereço do módulo de expansão analógico:

Início do Endereço da Entrada analógico = AIW (Extension Address ×10+10)

Início do Endereço da Saída analógico = AQW (Extension Address ×10+10)

Até 16 módulos de extensão podem ser conectados.

11.6. Define o endereço da extensão com um interruptor de discagem



Endereço do módulo de expansão = o valor do switch + 1

Switch		Valor
1	→	1
2	→	2
3	→	4
4	→	8

O valor do switch				Endereço do módulo de expansão
1	2	3	4	
OFF	OFF	OFF	OFF	1
ON	OFF	OFF	OFF	2
OFF	ON	OFF	OFF	3
OFF	OFF	ON	OFF	5
OFF	OFF	OFF	ON	9
ON	ON	OFF	OFF	4
ON	OFF	ON	OFF	6
ON	OFF	OFF	ON	10
OFF	ON	OFF	ON	11
OFF	OFF	ON	ON	13
OFF	ON	ON	OFF	7
ON	ON	ON	OFF	8
ON	ON	OFF	ON	12
ON	OFF	ON	ON	14
OFF	ON	ON	ON	15
ON	ON	ON	ON	16

11.7. Intervalo do endereço de host do CLP

Entrada Digital : I0.0~I1.7

Saída Digital : Q0.0~Q1.7

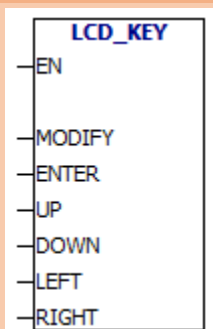
Entrada Analógica : AIW0~AIW18

Saída Analógica : AQW0~AQW18

12. Exemplos

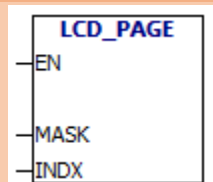
12.1. Exemplo instrução LCD

LCD_KEY



EN: Habilita
MODIFY: Modifica variáveis correspondentes.
ENTER: Confirma variáveis correspondentes
UP: Variável correspondente ao botão UP.
DOWN: Variável correspondente ao botão DOWN.
LEFT: Variável correspondente ao botão LEFT.
RIGHT: Variável correspondente ao botão RIGHT.

LCD_PAGE



LCD_PAGE: vincula as instruções do LCD exibidas no display.
EN: Habilitado
MASK: A máscara atual do grupo de páginas, é 1.
INDX: Número da página atual. Você pode modificar o número da página, o LCD exibirá a página.

Complementar: A entrada MASK é um byte.

Tome VB0 como um Exemplo:



Quando o bit 0 é igual a 1, o LCD exibirá 0 grupo.

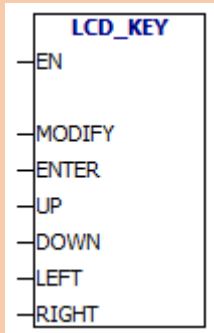
Quando o bit 1 é igual a 1, o LCD exibirá 1 grupo.

Quando o bit 2 é igual a 1, o LCD exibirá 2 grupo.

·
·

Quando o bit 7 é igual a 1, o LCD exibirá 7 grupo.

LCD_EDIT



LCD_EDIT: vincula as variáveis do CLP para editar o estado do LCD.

EN: Habilita

UNIT: Edita o número dos objetos na página.

AUTO: Usa as teclas de LCD para editar.

DEPTH: A edição profunda no objeto de edição.

LOOP: editar LOOP.

FLASH: O objeto de edição está piscando ou não.

V: O valor atual do objeto de edição.

VMIN: O valor mínimo do objeto de edição.

VMAX: O valor máximo do objeto de edição.

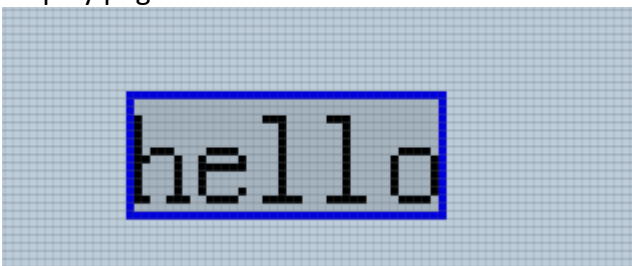
Exemplo :

Você deve editar as páginas de exibição no software do LCD.

Display página 1:



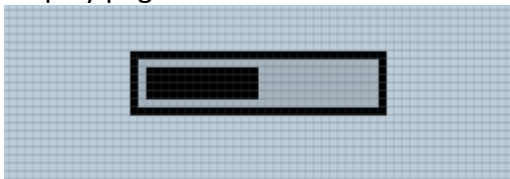
Display página 2:



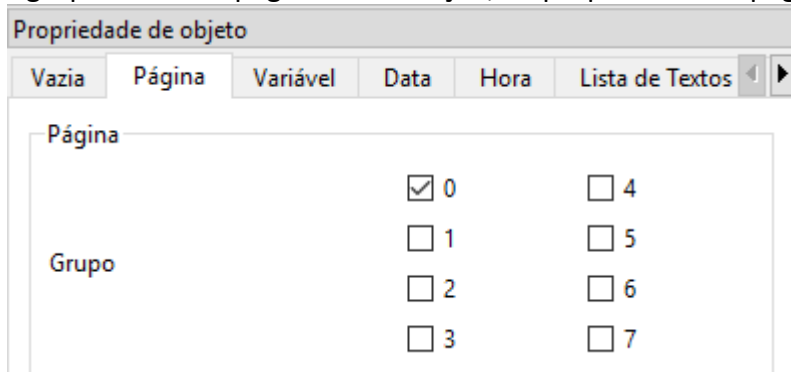
Display página 3:



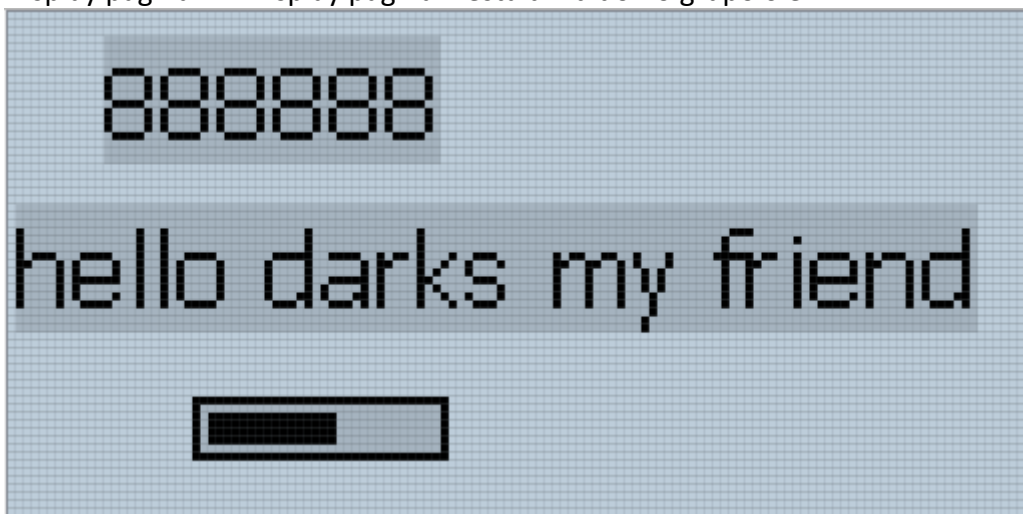
Display página 4:



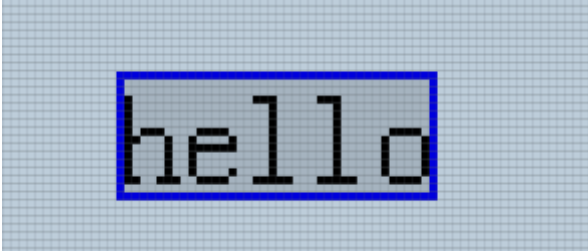
Agrupamento de páginas de exibição, na propriedade da página de exibição:



Display página 1 : Display página 1 está dividido no grupo 0 e 1.



Display página 2 : Display página 2 está dividido no grupo 0 e 1.



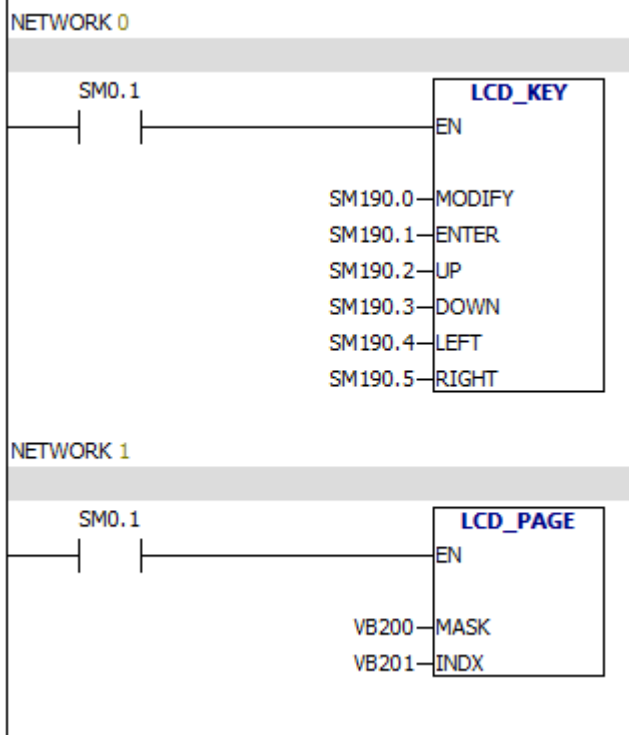
Display página 3 : Display página 3 está no grupo 0.

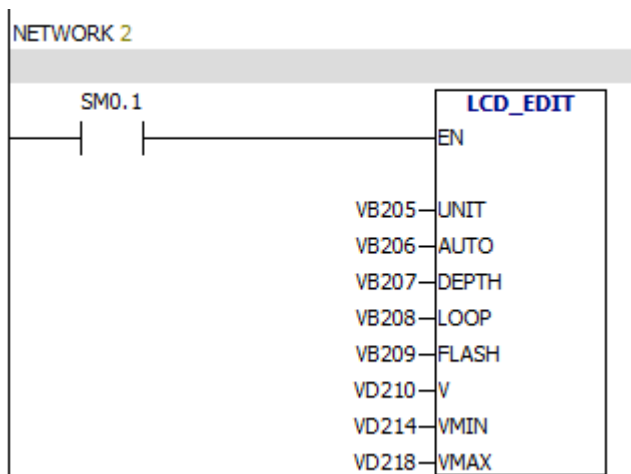


Display página 4 : Display página 4 está no grupo 0.



Programa do CLP:





Análise:

Na Network 0, o programa liga as chaves LCD e as variáveis do CLP.

O CLP possui dez teclas de função. Cada tecla de função corresponde a uma variável de CLP.

F1 corresponde à SM191.0

F2 corresponde à SM191.1

F3 corresponde à SM191.2

F4 corresponde à SM191.3

ESC corresponde à SM190.0

OK corresponde à SM190.1

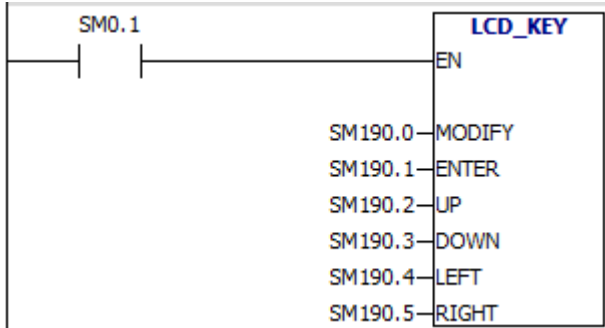
UP corresponde à SM190.2

DOWN corresponde à SM190.3

LEFT corresponde à SM190.4

RIGHT corresponde à SM190.5

NETWORK 0:



MODIFY é a tecla de função ESC, corresponde a SM190.0

As funções das teclas de função:

Você pode personalizar F1 ~ F4.

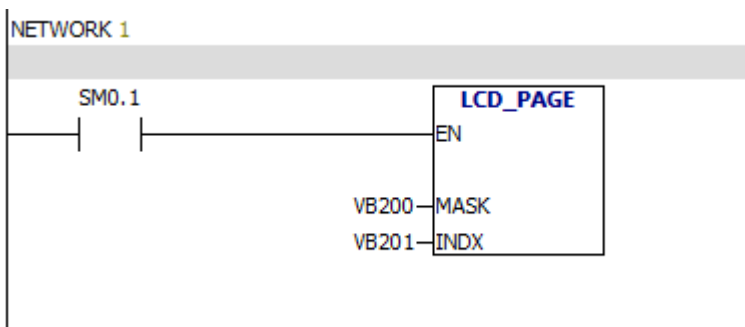
ESC é usado para modificar valores e sair.

OK é usado para confirmar valores.

As teclas de função UP e DOWN podem alternar a página de exibição. Eles também podem aumentar ou diminuir os valores.

As teclas de função LEFT e RIGHT podem ser usadas para trocar objetos de edição.

A função da NETWORK 1 é uma variável do CLP vinculativa às páginas do LCD.



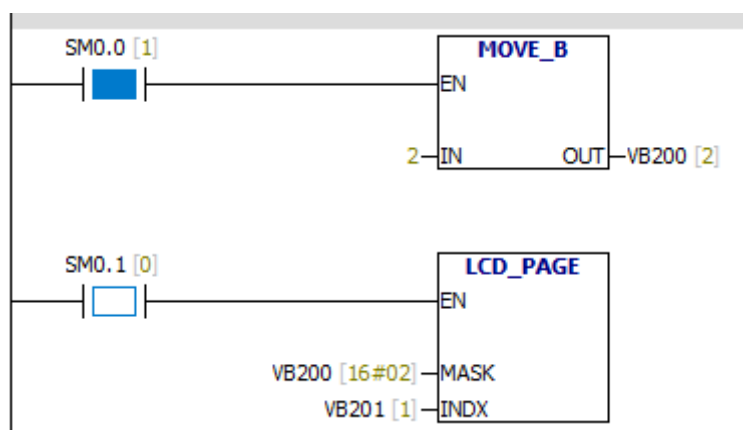
Resultados de Operação:

Address	Data Type	Value
VB200	BYTE	16#01
VB201	SINT	0

O Display da página 1 do grupo 0 é exibida por padrão. O valor de VB200 é de 1, 0-bit é igual a 1, então o LCD exibe 0 grupo. O valor de VB201 é 0, o que significa a primeira página de exibição. A primeira página de exibição é o Display da página 1.

Você pode usar o programa para especificar o grupo de exibição e a página de exibição.

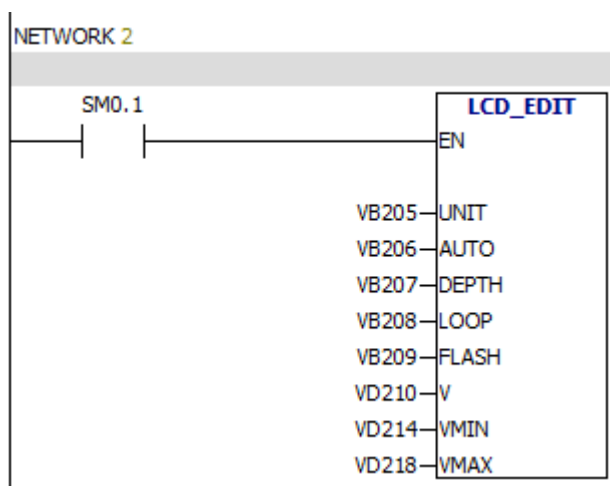
Exemplo:



O LCD exibirá o grupo 1 no Display da página 2.

Se você usar as teclas de função LCD para alternar as páginas de exibição, o valor de VB201 mudará.

NETWORK2:



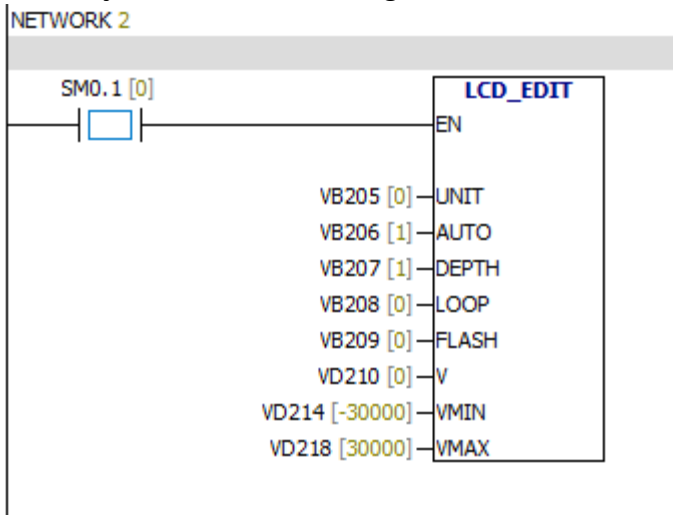
LCD -EDIT a instrução liga as variáveis do PLC e os estados de edição do LCD.

Exemplo:

Quando você modifica a primeira variável do Display da página 1:



A instrução será exibida da seguinte forma:



Vb205 = 0 Variável 0, é a primeira variável.

Vb206 = 1 Isso significa que você pode usar as teclas de função do LCD para editar variáveis.

Vb207 = 1 Isso significa que você pode modificar um único dígito. Vb207 = 2, você pode modificar o dígito de dezenas de dígitos.

Vb208 = 0 Sem loop

Vb209 = 0 Sem flicker

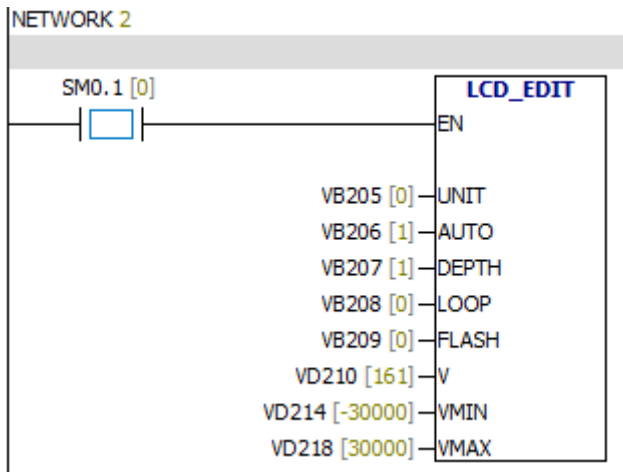
Vd210 = 0 O valor atual da variável é 0

Vd214 = -30000 O valor mínimo é -30000

Vd218 = 30000 O valor máximo é 30000

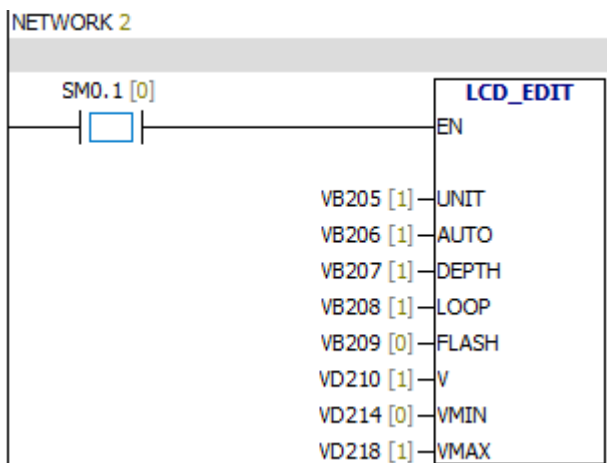
Exemplo:

Modifique o valor da variável para 161.



Exemplo:

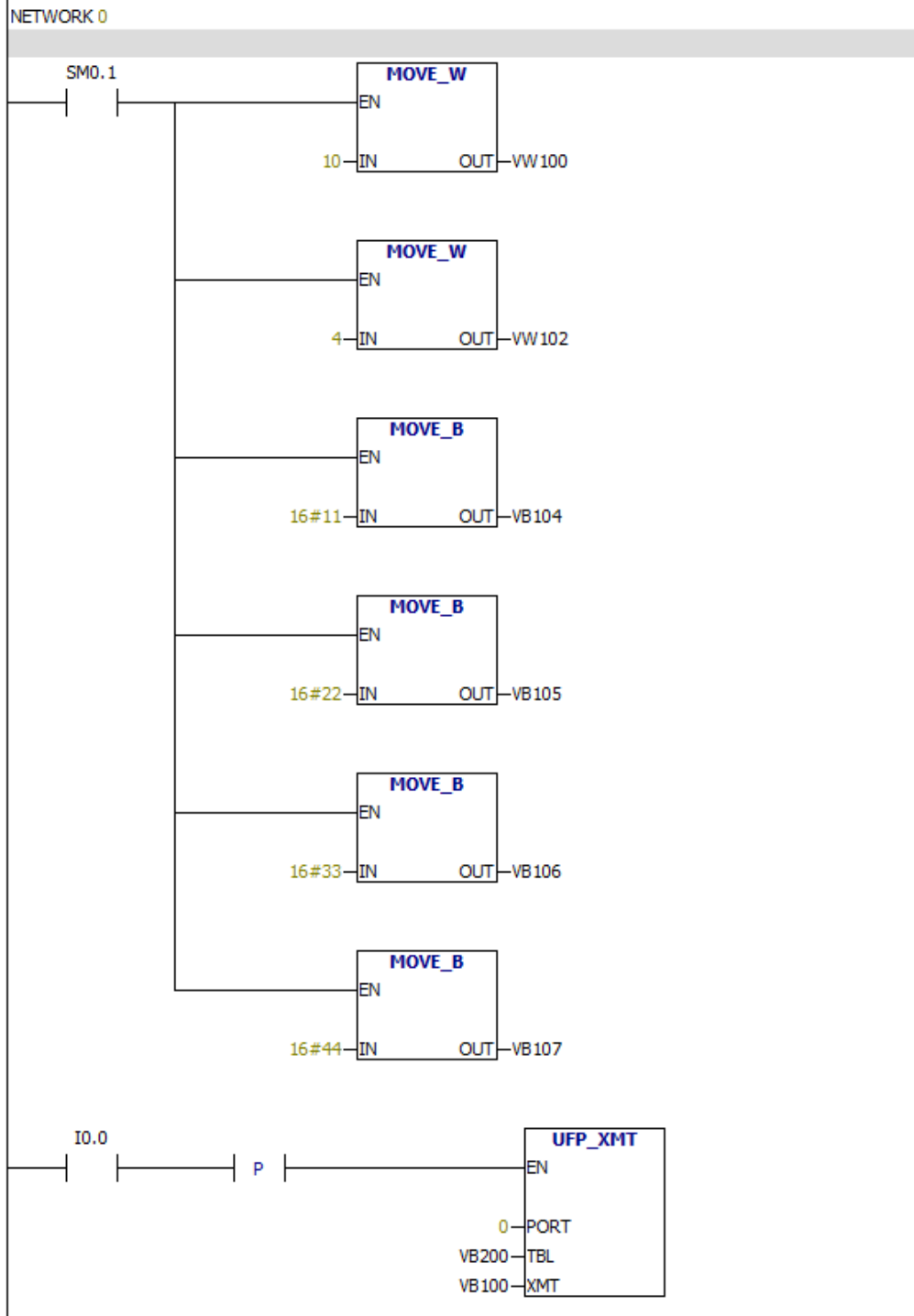
Modifique lista de texto



- VB205 Variável 1, é a segunda variável.
- VB206 Isso significa que você pode usar as teclas de função LCD para editar variáveis.
- VB207 A profundidade de edição é 1.
- VB208 LOOP
- VB209 Sem flicker
- VD210 O valor atual da variável é 1
- VD214 O valor mínimo é 0
- VD218 O valor máximo é 1

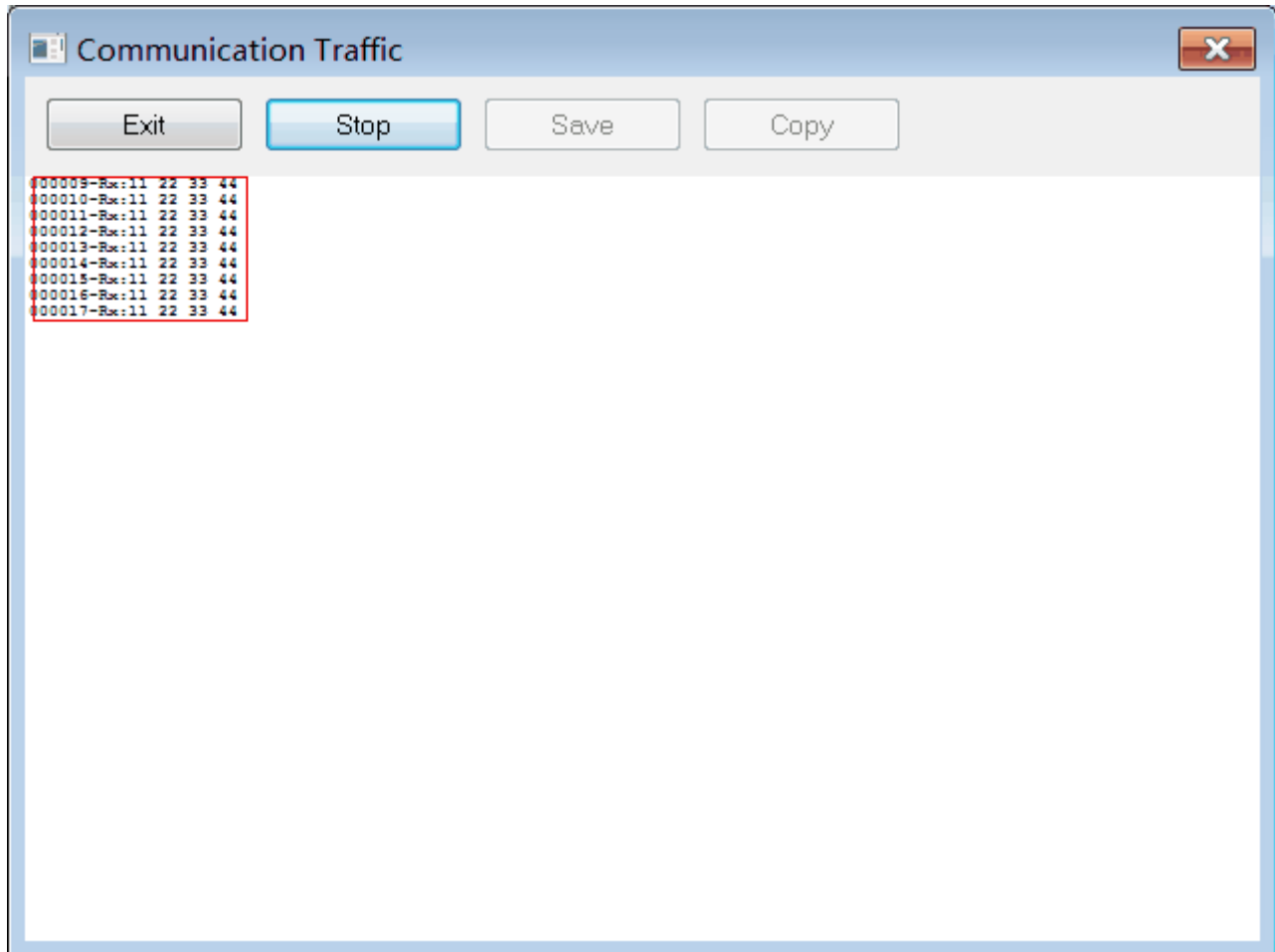
12.2. Porta serial porta livre

Programa 1

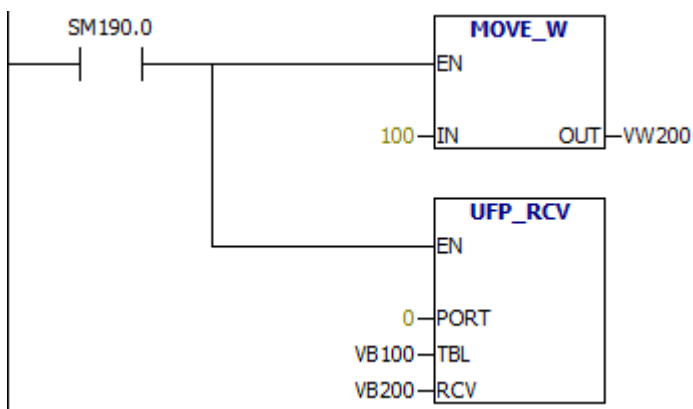


Quando o valor de I0.0 é igual a 1, o PLC envia dados "16 # 11 16 # 22 16 # 33 16 # 44" para a estação escrava através da porta 0.

Dados recebidos pela estação escrava:

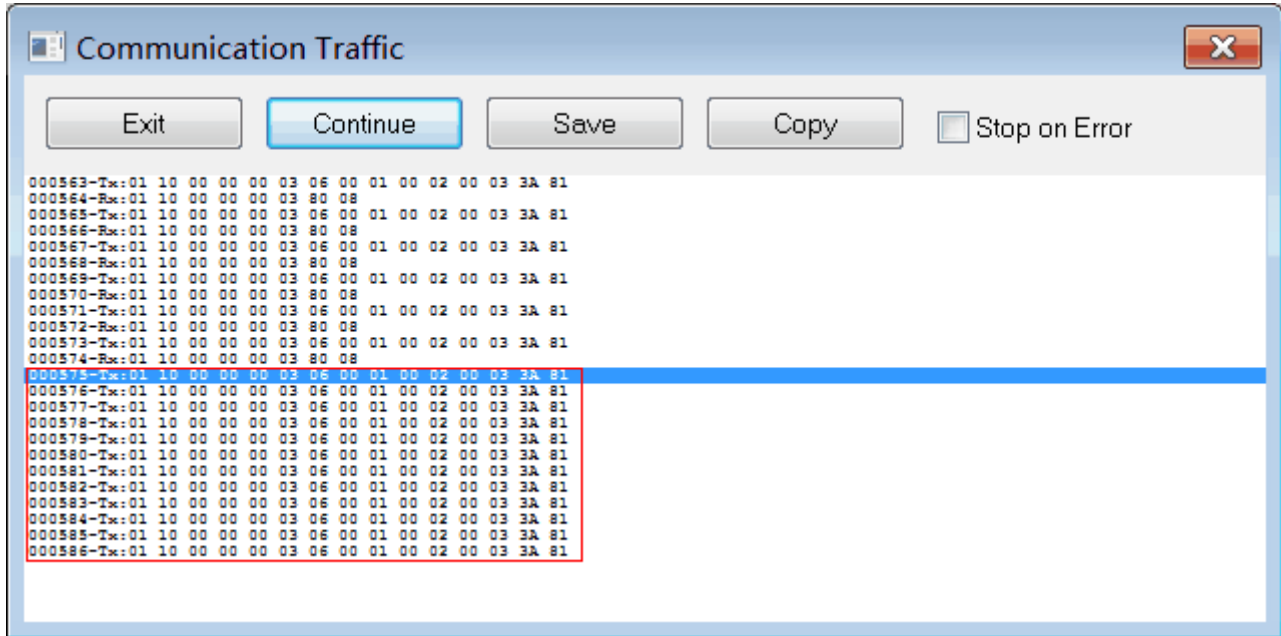


Programa 2



Quando o valor de SM190.0 é igual a 1, o CLP recebe dados da estação mestre através da porta 0. E coloque os dados na tabela VW200.

Dados enviados pela estação principal:



Dado recebido via CLP

Status Chart

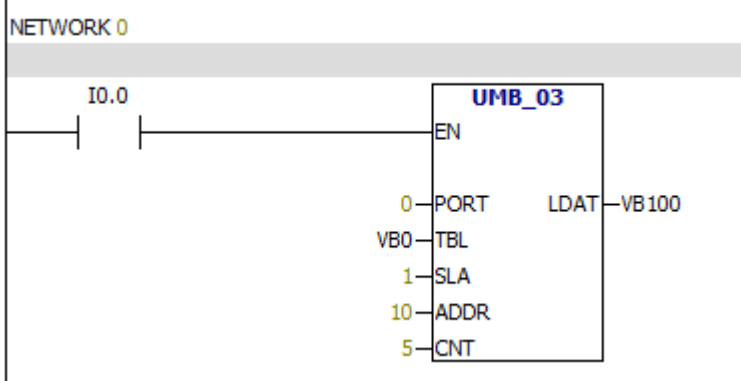
Address	Data Ty...	Value
VW200	INT	100
VW202	INT	15
VB204	SINT	1
VB205	SINT	16
VB206	SINT	0
VB207	SINT	0
VB208	SINT	0
VB209	SINT	3
VB210	SINT	6
VB211	SINT	0
VB212	SINT	1
VB213	SINT	0
VB214	SINT	2

O resultado está correto.

12.3. Comunicação MODBUS

Programa 1

Estação Master



Lê 5 words da estação Slave.

Estação Slave

Modbus Slave - [Mbslav1]

File Edit Connection Setup Display View Window

ID = 1: F = 03

	Alias	00010
0		1
1		2
2		3
3		4
4		5
5		6
6		7
7		8
8		9
9		10

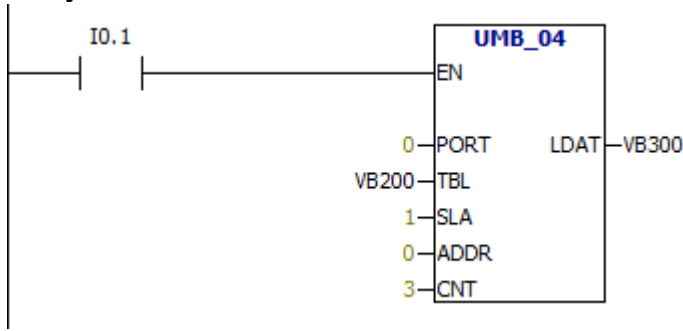
For Help, press F1.

Valor lido pela estação principal:

Status Chart			
	Address	Data Ty...	Value
	VW100	INT	1
	VW102	INT	2
	VW104	INT	3
	VW106	INT	4
	VW108	INT	5

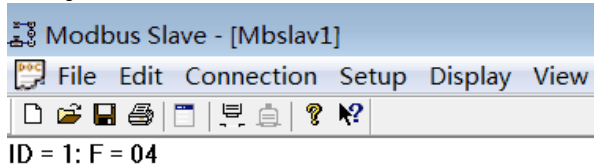
Programa 2

Estação Master



Lê o valor dos 3 analógicos

Estação Slave



	Alias	00000
0		123
1		456
2		789
3		0
4		0
5		0
6		0
7		0
8		0
9		0

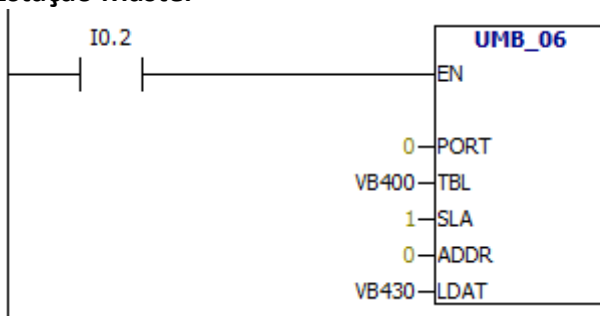
For Help, press F1.

O valor lido pela estação principal

Status Chart			
	Address	Data Ty...	Value
	VW300	INT	123
	VW302	INT	456
	VW304	INT	789

Programa 3

Estação Master



O valor de VW430 =1000

Estação Slave

Modbus Slave - [Mbslav1]

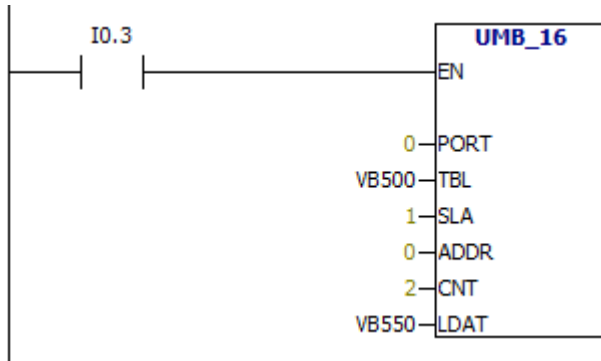
File Edit Connection Setup Display

ID = 1: F = 03

	Alias	00000
0		1000
1		0
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

Programa 4

Estação Master



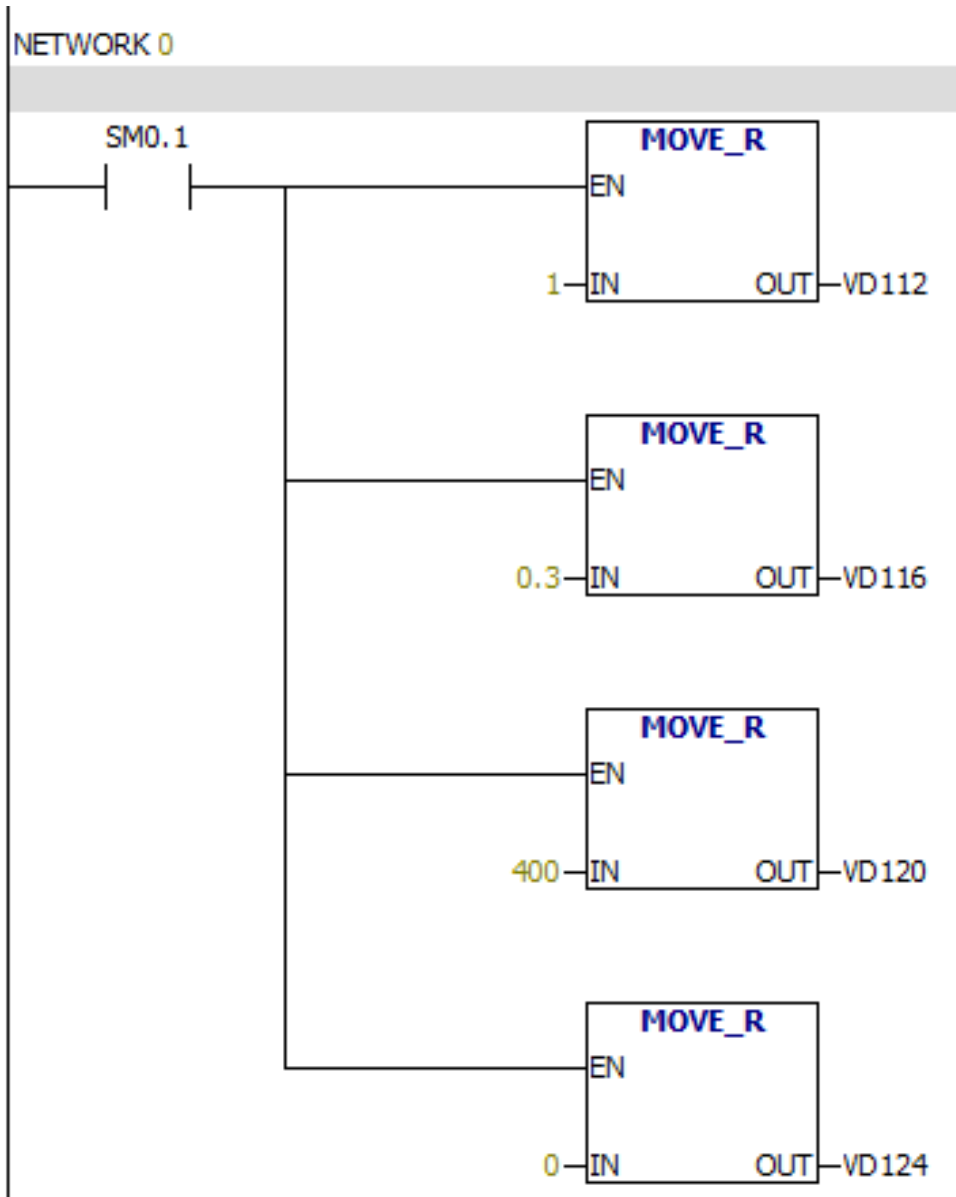
O valor de VW550=888

O valor de VW552=999

Estação Slave

	Alias	00000
0		888
1		999
2		0
3		0
4		0
5		0
6		0
7		0
8		0
9		0

12.4. Exemplo da instrução PID



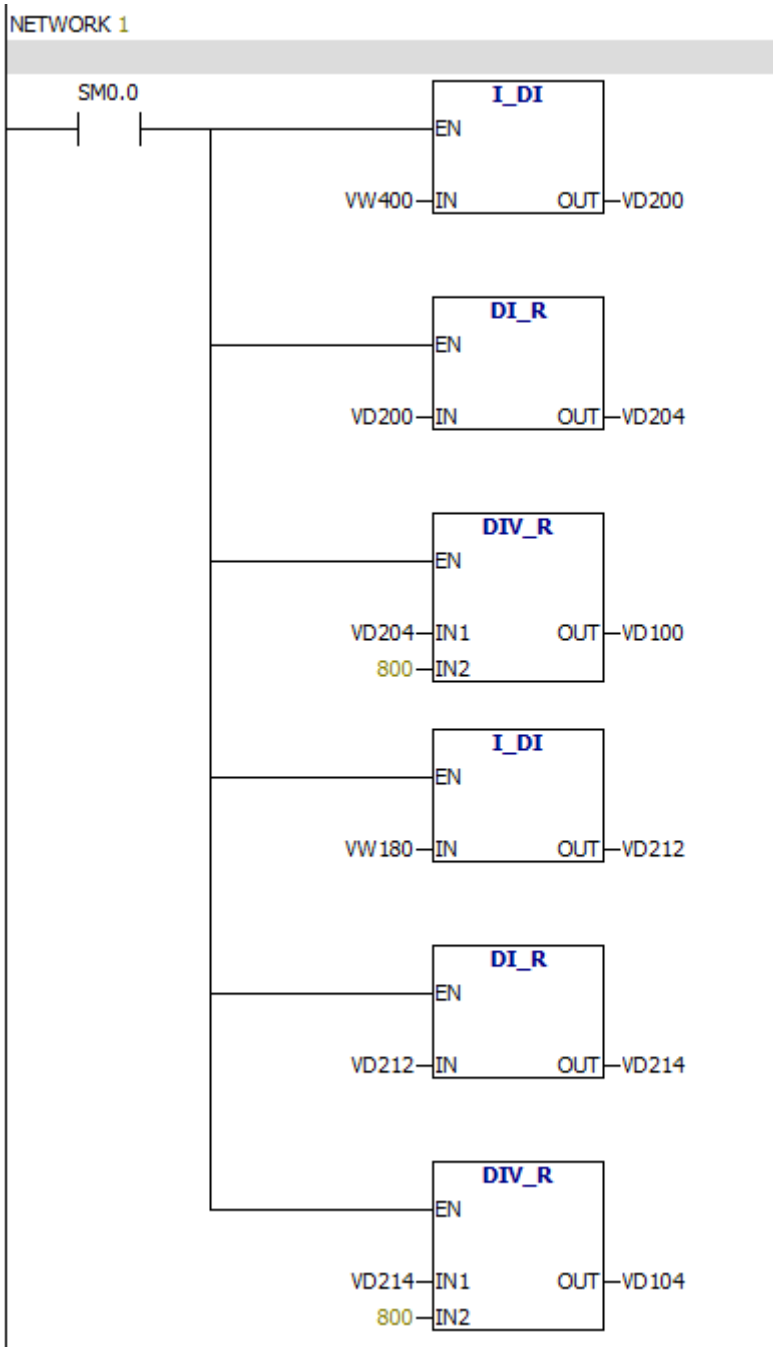
O parâmetros iniciais do PID

VD112 Ganho

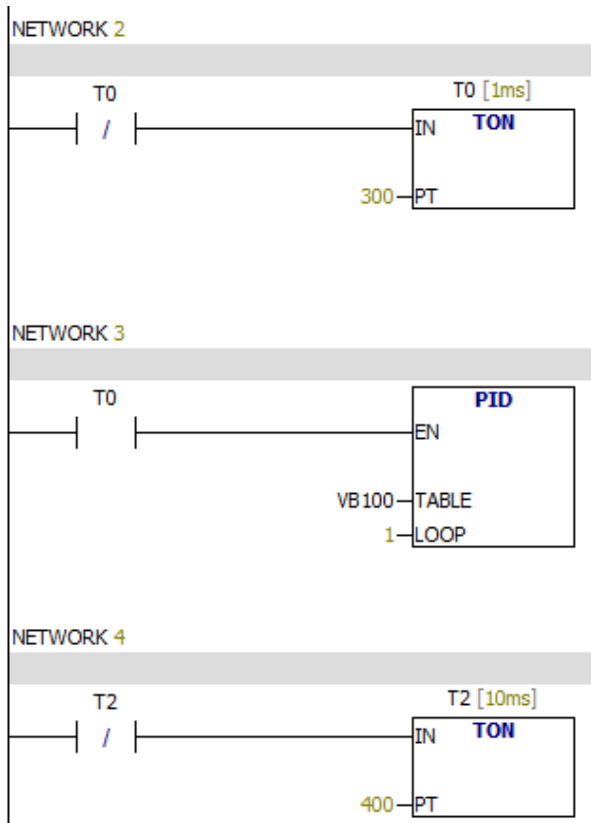
VD116 Tempo de amostragem

VD120 Tempo de integração

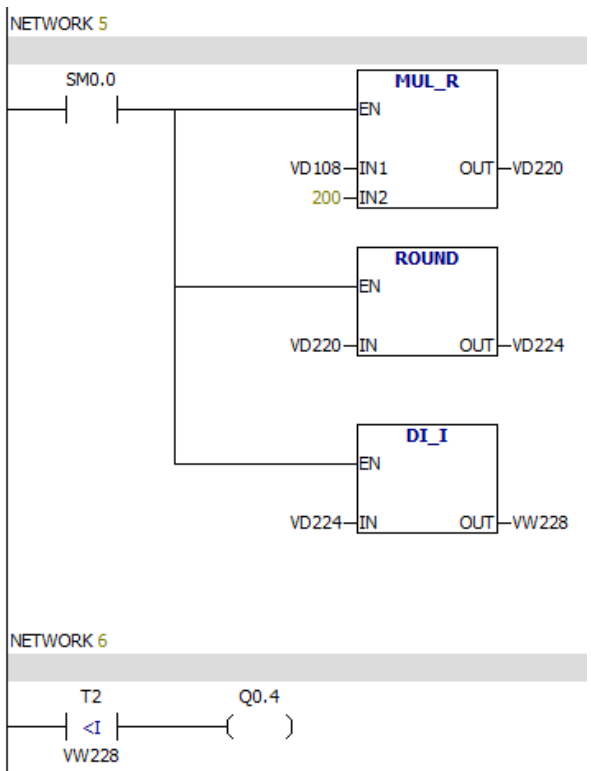
VD124 Tempo de diferencial



Conversão da quantidade do processo e a unidade de valor ajustado



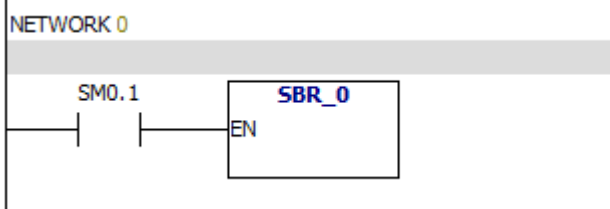
Chama o comando PID a cada 0,3 segundos.



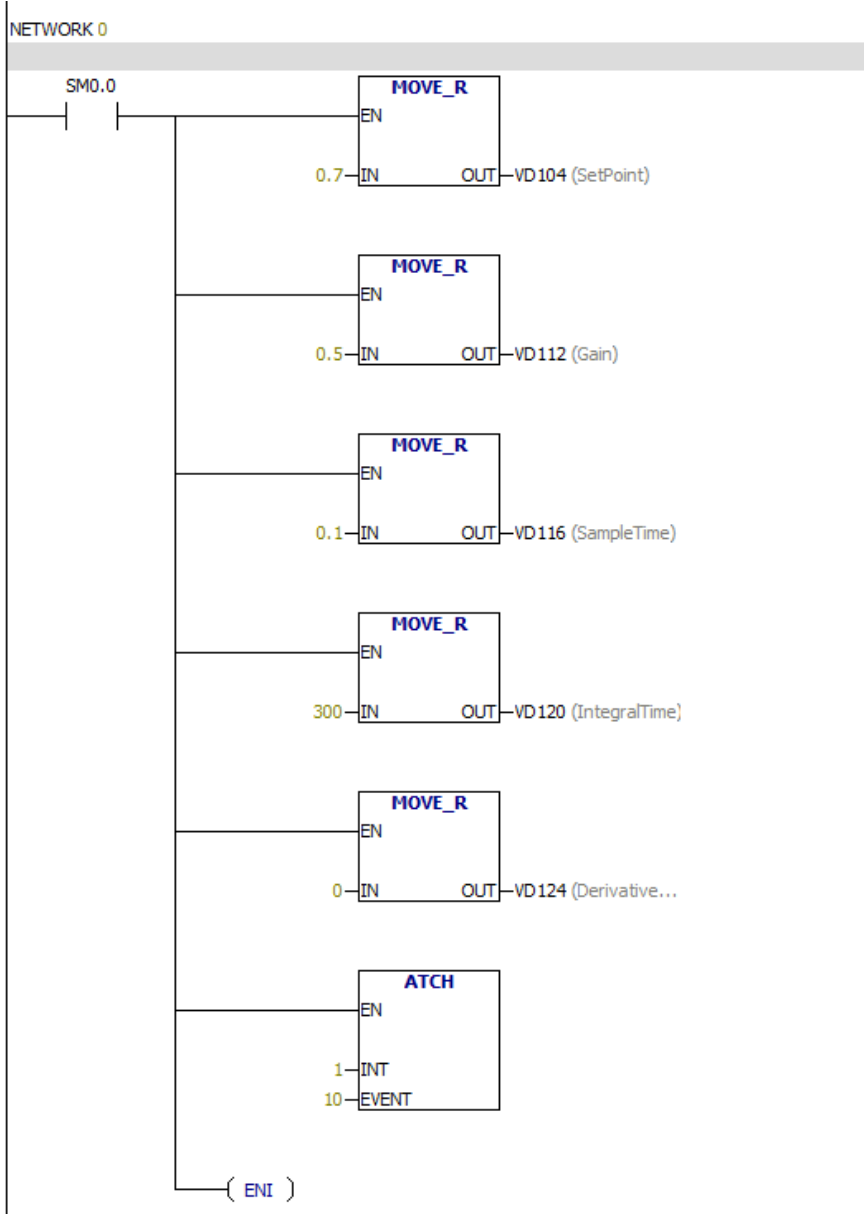
A conversão do valor de saída.

12.5. PID example 2

Programa principal:

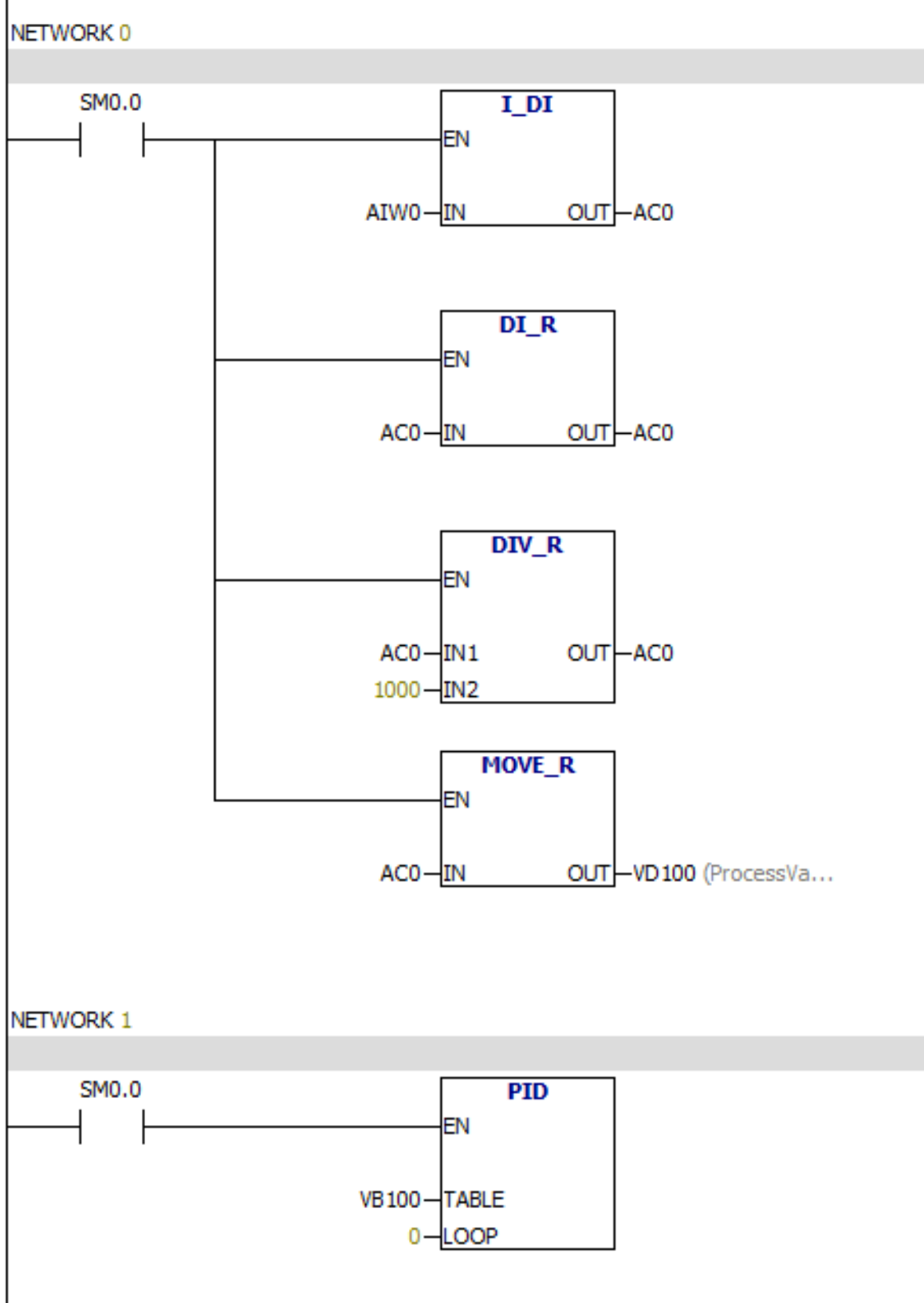


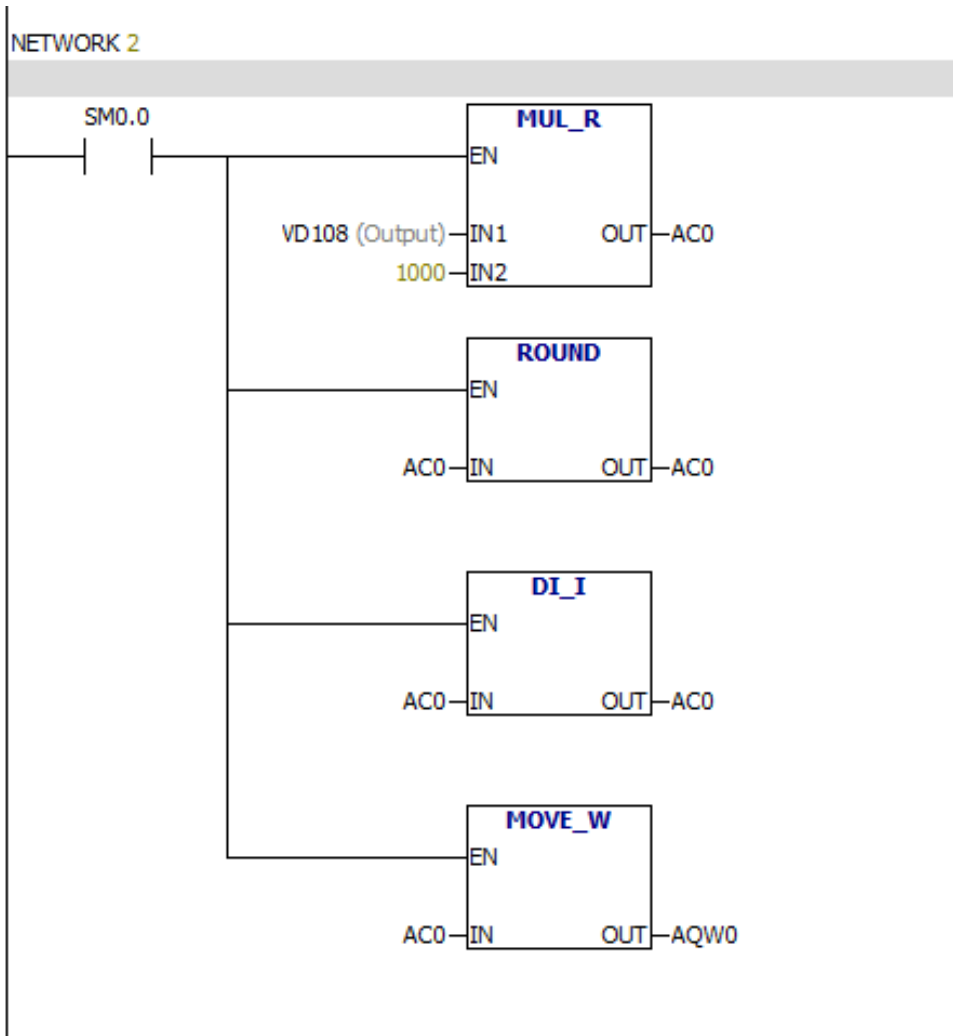
Chama sub-rotina



Define parâmetros PID, interrupção de tempo de conexão

Interrupção de programa:





Converte o valor da variável de processo, executa a instrução PID, converte o valor de "OUT".

Símbolo de variável:

	Symbol	Adress	Data Ty...	Comment
✓	ProcessVariable	VD100	REAL	
✓	SetPoint	VD104	REAL	
✓	Output	VD108	REAL	
✓	Gain	VD112	REAL	
✓	SampleTime	VD116	REAL	
✓	IntegralTime	VD120	REAL	
✓	DerivativeTime	VD124	REAL	

Configuração de tempo de interrupção:

The screenshot shows a software window titled "Bloco do Sistema" with a close button (X) in the top right corner. The window contains a tabbed interface with four tabs: "RS232/RS485", "RS232/RS485", "CAN", and "Senha". The "RS232/RS485" tab is selected, and it is divided into three sub-sections: "Memória Retentiva", "Interrupção", and "Forçar Saídas". The "Interrupção" sub-section is active and contains two input fields for interrupt timing, both set to "200". The first field is labeled "Tempo Interrup 0 (1ms)" and the second is "Tempo Interrup 1 (1ms)". Both fields have a range indicator "(Intervalo 1 ... 255)". A "Padrão" button is located in the "Forçar Saídas" sub-section. At the bottom of the window, there is a message: "As modificações só terão efeito após o Download". At the very bottom, there are "OK" and "Cancelar" buttons.

RS232/RS485	RS232/RS485	CAN	Senha
Memória Retentiva	Interrupção		Forçar Saídas

Tempo Interrup 0 (1ms) (Intervalo 1 ... 255)

Tempo Interrup 1 (1ms) (Intervalo 1 ... 255)

Padrão

As modificações só terão efeito após o Download

OK Cancelar

Ponto de ajuste: 0,7

Saída: AQW0

Ganho: 0,5

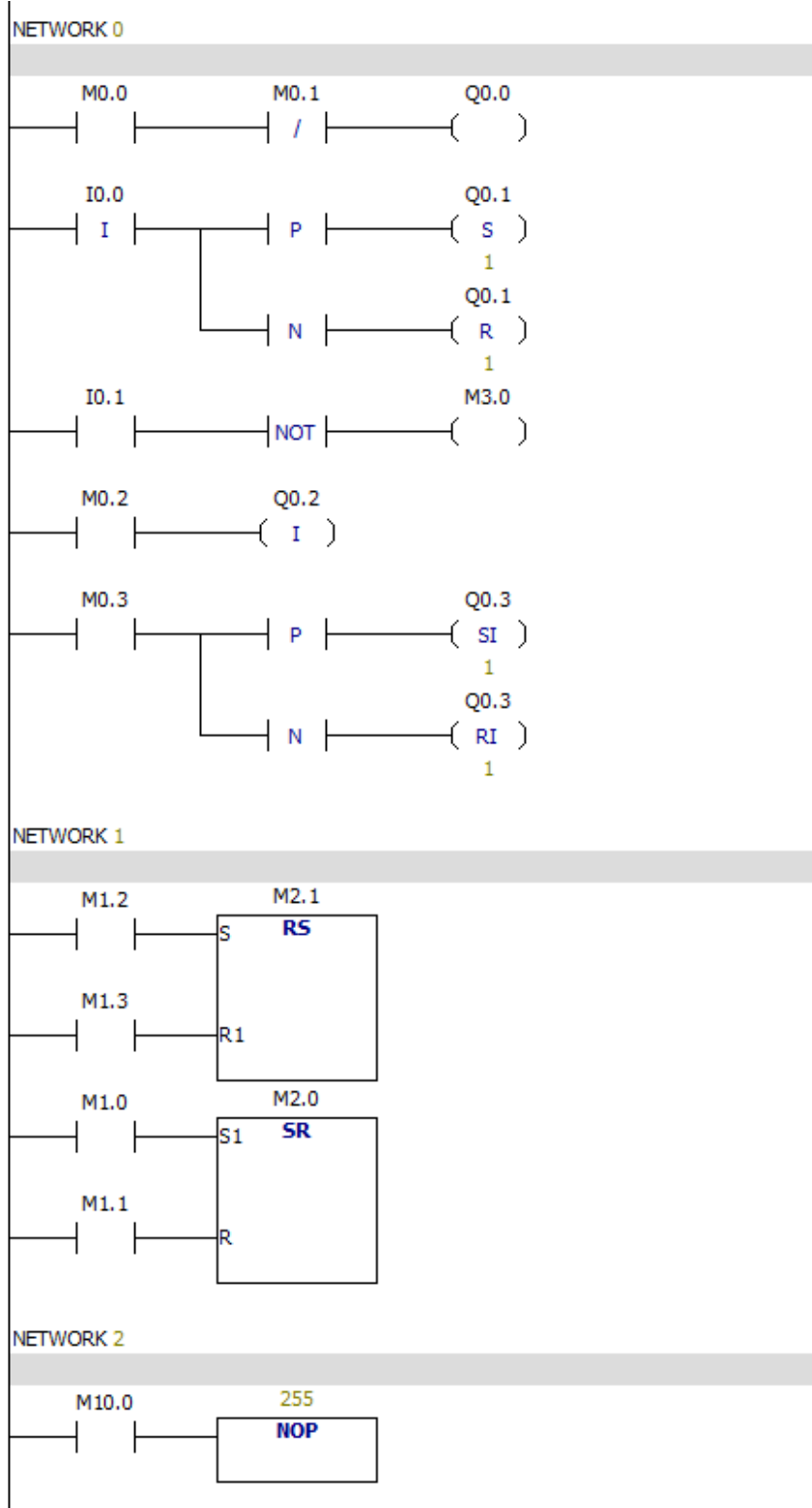
Tempo de amostragem: 0,1

Tempo integral: 200

Tempo derivado: 200

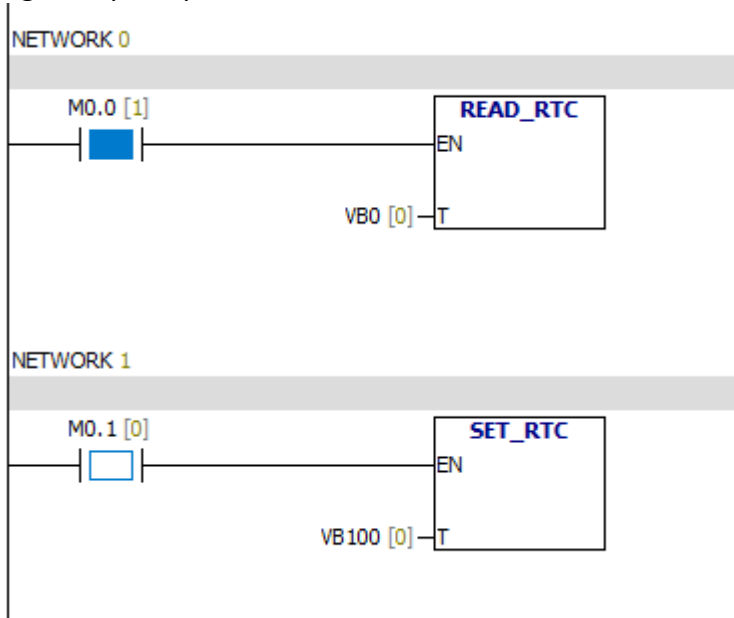
12.6. Exemplo de Bit Lógico

Exemplo de Bit Lógico:



12.7. Exemplo de instrução Clock

Programa principal:



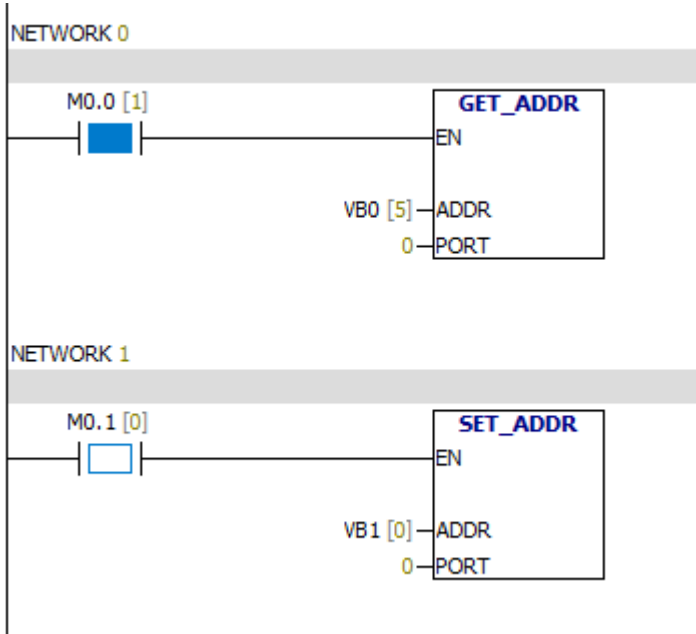
Status chart:

Status Chart				
	Address	Data Ty...	Value	Forced
	VB0	SINT	2	
	VB1	SINT	30	
	VB2	SINT	10	
	VB3	SINT	6	
	VB4	SINT	1	
	VB5	SINT	12	
	VB6	SINT	16	
	VB7	SINT	0	
	M0.0	BOOL	1	
	M0.1	BOOL	0	

VB0 2 Segundo
 VB1 30 Minuto
 VB2 10 Hora
 VB3 6 Data
 VB4 1 Semana
 VB5 12 Mês
 VB6 16 Ano

12.8. Exemplo de instrução de comunicação

Programa principal:



Status Chart:

Status Chart				
	Address	Data Ty...	Value	Forced
	M0.0	BOOL	1	
	M0.1	BOOL	0	
	VB0	SINT	5	
	VB1	SINT	0	

Bloco do Sistema:

Bloco do Sistema

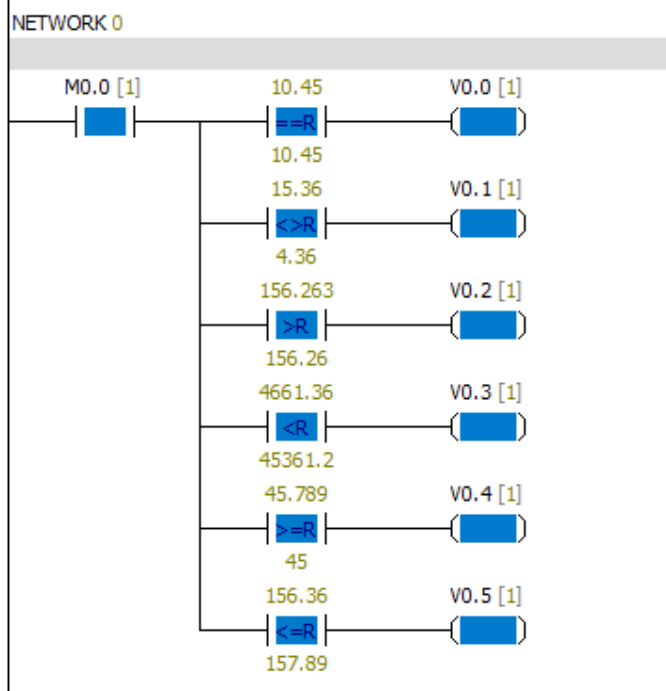
Memória Retentiva	Interrupção		Forçar Saídas
RS232/RS485	RS232/RS485	CAN	Senha
Padrão			
	Porta 0	Porta 1	
Protocolo:	Modbus	Modbus	
Estação:	1	1	(Intervalo 0... 255)
Velocidade:	9600 bps	9600 bps	
Bit dados:	8 (RTU)	8 (RTU)	
Paridade:	NONE	NONE	
b.Parada:	1 Bit	1 Bit	
Tempo Resposta (100ms)	10	10	(Intervalo 1... 255)
Intervalo de quadros (B)	10	10	(Intervalo 1... 255)

As modificações só terão efeito após o Download

OK Cancelar

12.9. Exemplo de comparação de número Real

Programa principal:



12.10.Exemplo de comparação de String

Programa principal:

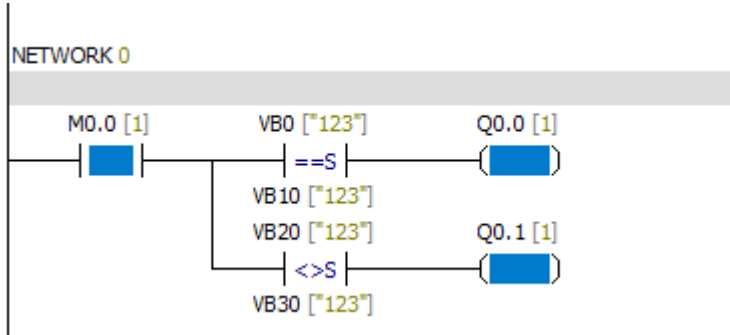


Tabela de Monitoramento:

Status Chart								
Address	Data Ty...	Value	Forced	Address	Data Ty...	Value	Forced	
M0.0	BOOL	1		VB20	SINT	3		
				VB21	SINT	49		
VB0	SINT	3		VB22	SINT	50		
VB1	SINT	49		VB23	SINT	51		
VB2	SINT	50						
VB3	SINT	51						
VB10	SINT	3		VB30	SINT	4		
VB11	SINT	49		VB31	SINT	49		
VB12	SINT	50		VB32	SINT	50		
VB13	SINT	51		VB33	SINT	51		
				VB34	SINT	0		

Trabalho elaborado e editado pelo Departamento de Engenharia de Aplicação da Metaltex

Manual de Instalação e Operação – Metaltex.

Núcleo de estudos Novembro/2017.

- Coordenação do Projeto:
- Elaboração:
- Revisão:

Maike Ramos de Jesus
Maike Ramos de Jesus
Gilberto Jana

- Revisão Técnica Prática:

-E-mail:

mjesus@metaltex.com

Manual de Instalação e Operação – Metaltex.



Introdução

Configuração

Aplicações

NEXO MANUAL DO LCD INCORPORADO + Guia de Ferramentas

Contents

Instruções para uso do LCD NeXo.....	3
Introdução	3
Interface.....	4
1. Barra de Menus.....	5
1.1 Arquivo.....	5
1.3 Editar.....	13
1.4 Ferramenta	13
1.5 Ajuda	14
2. Barra de Ferramentas	15
2.1 Variável	15
2.2 Data.....	18
2.3 Hora	20
2.4 Lista de Textos	21
2.5 Lista de Imagens	24
2.6 Texto Estático.....	26
2.7 Texto Animado.....	27
2.8 Imagem Estática.....	30
2.9 Imagem Animada.....	31
2.10 Barra de Progresso Estática	34
2.11 Barra de Progresso Animada	35
3. Propriedades da Página	38
4. Variáveis do CLP	40

Instruções para uso do LCD NeXo



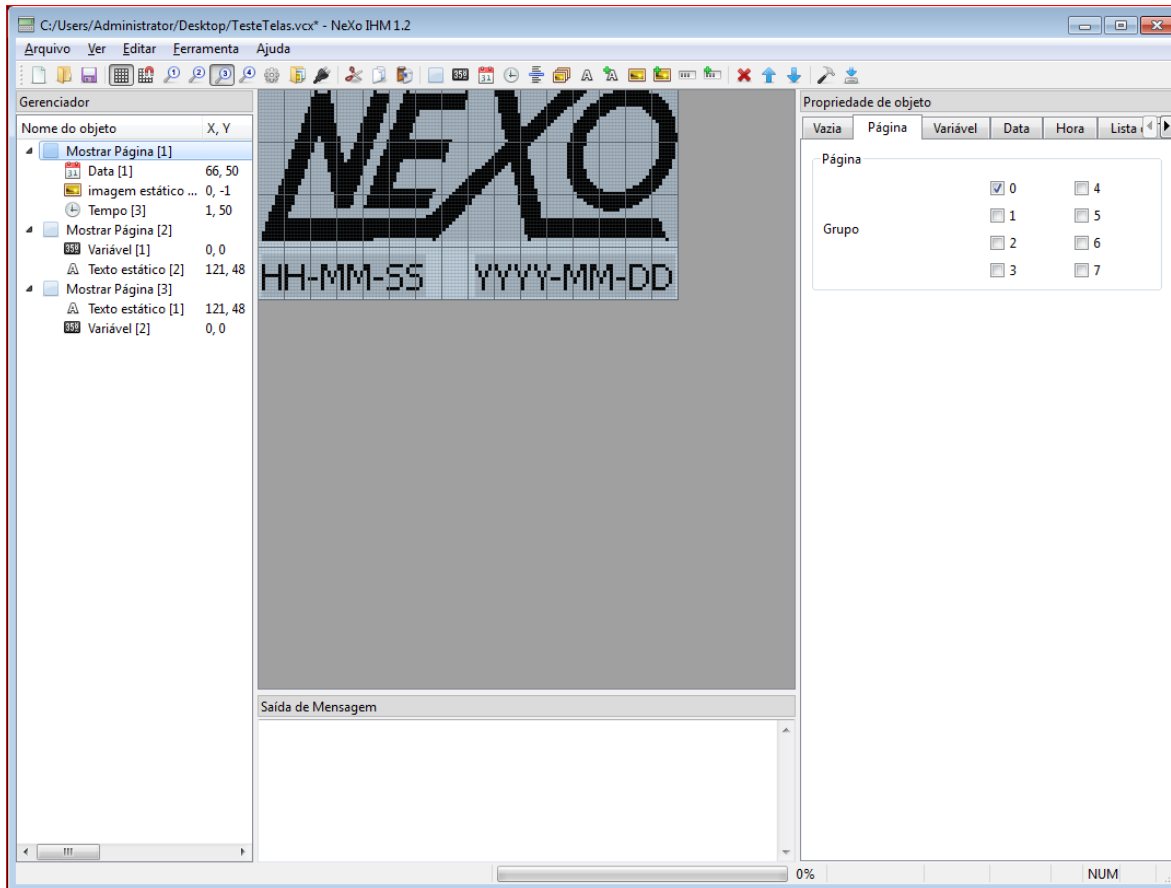
Introdução

Este documento é um auxílio para uso do LCD incorporado no Mini CLP NeXo.

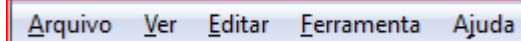
Suporte para os seguintes modelos de Mini CLP:

NEX14-AR
NEX14-DR
NEX18-AR
NEX18-DR
NEX18-DRT

Interface

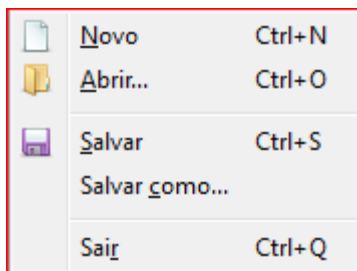


1. Barra de Menus



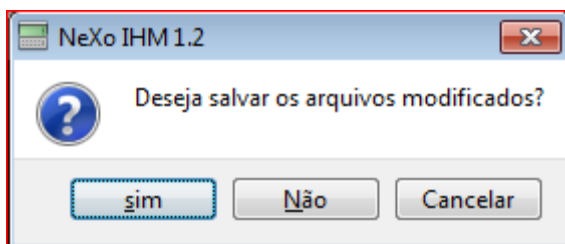
Arquivo Ver Editar Ferramenta Ajuda

1.1 Arquivo

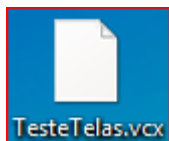


Novo: Criar um novo projeto.

Você receberá uma mensagem para confirmar se deseja salvar o projeto atual.



Abrir: Abrir um projeto salvo. O formato do arquivo é *.VCX

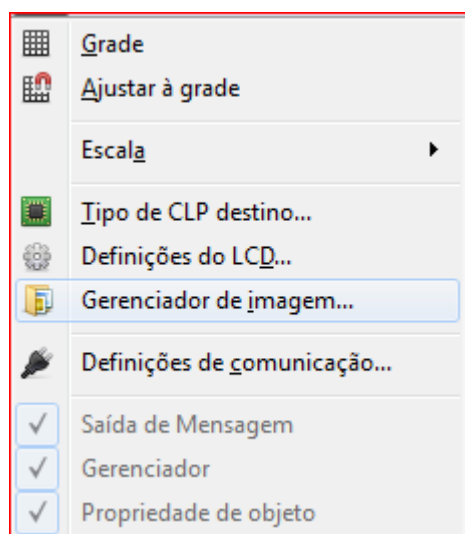


Salvar: Salva o projeto atual.

Salvar como: Salvar projeto com outro nome ou em outro local.

Sair: Sair do NeXo IHM.

1.2 Ver

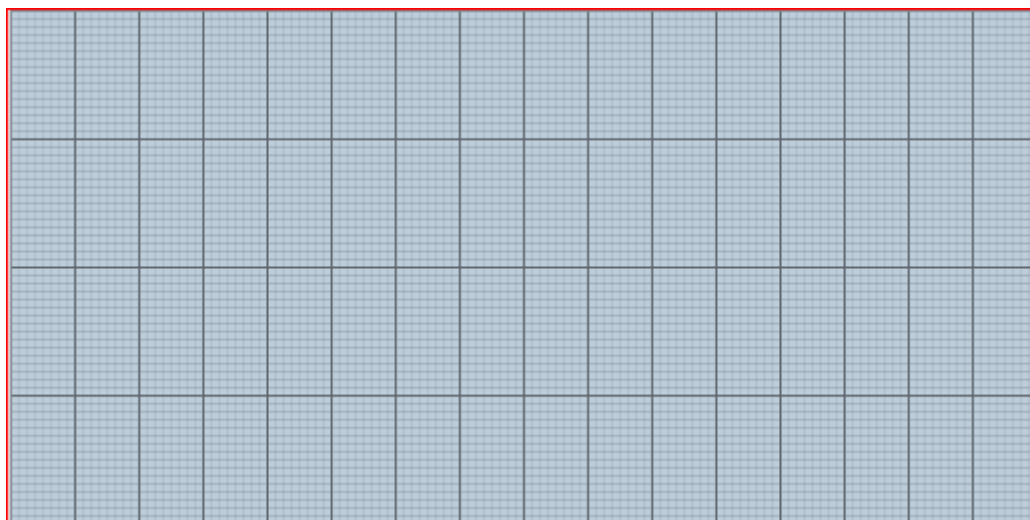


Grade: Adiciona uma grade a área de edição de tela para auxiliar o posicionamento e ajuste dos componentes.

Sem Grade:

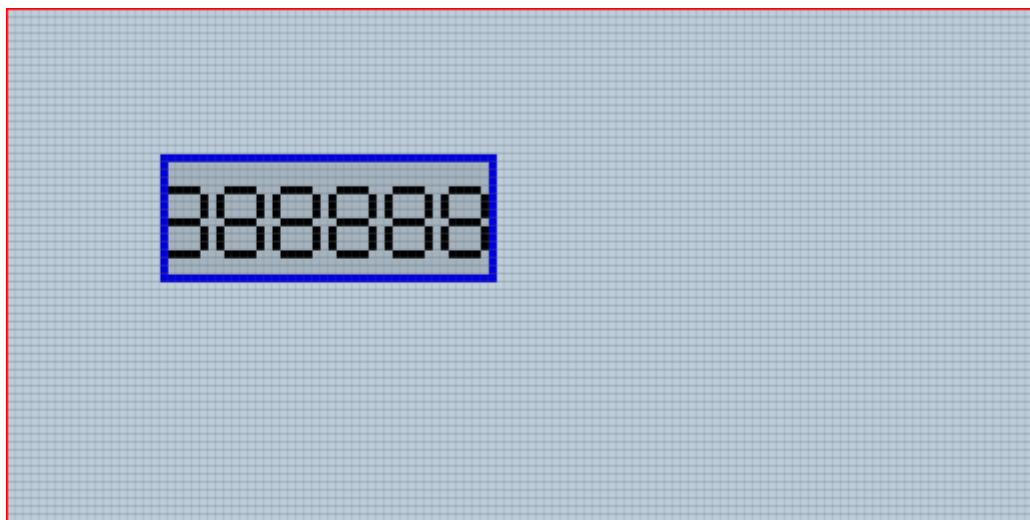


Com Grade

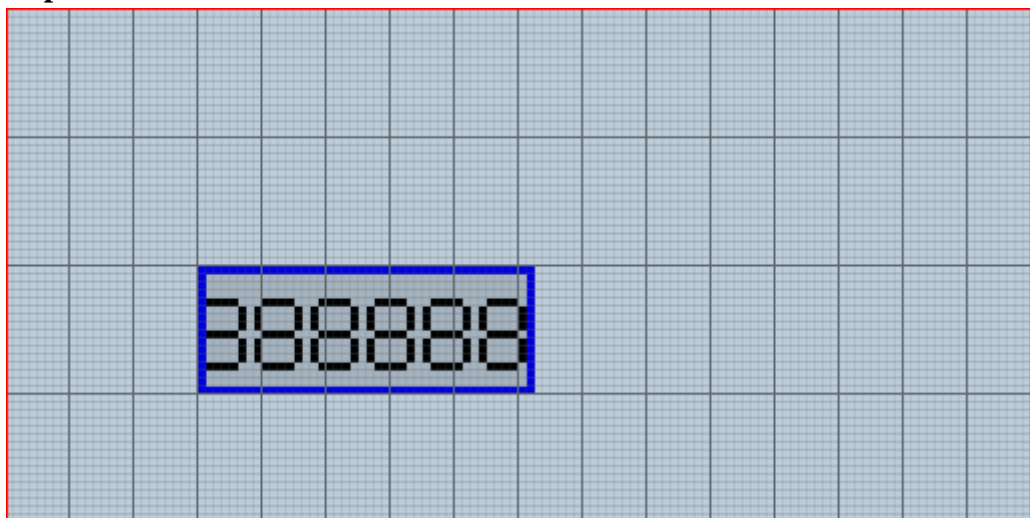


Ajustar à Grade : Alinha os elementos na Grade

Antes:



Depois:

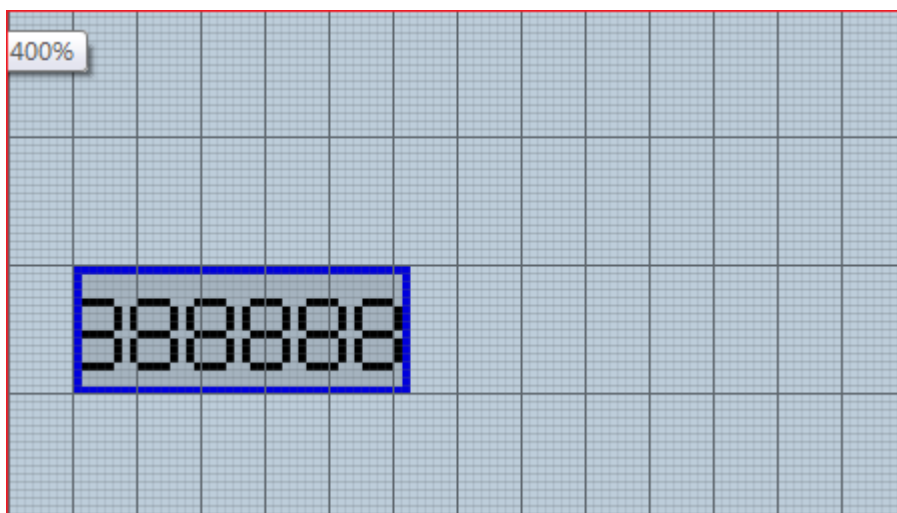
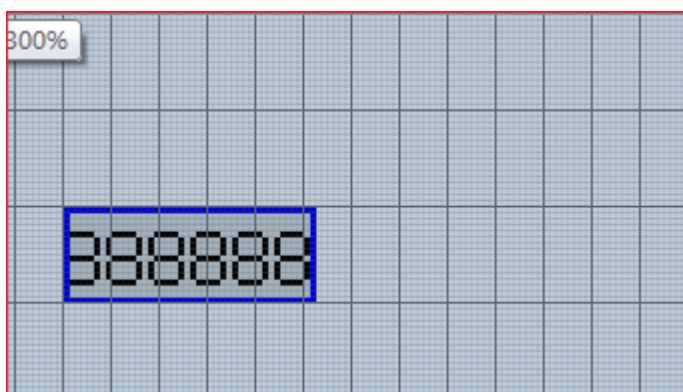
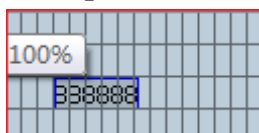


Escala: Ajusta o zoom da visualização da pagina:

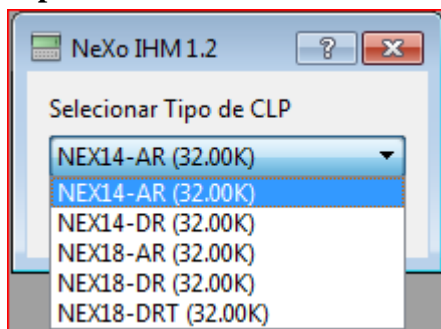
Escalas acessíveis:

- 1 [1] Escala 100%
- 2 [2] Escala 200%
- 3 [3] Escala 300%
- 4 [4] Escala 400%

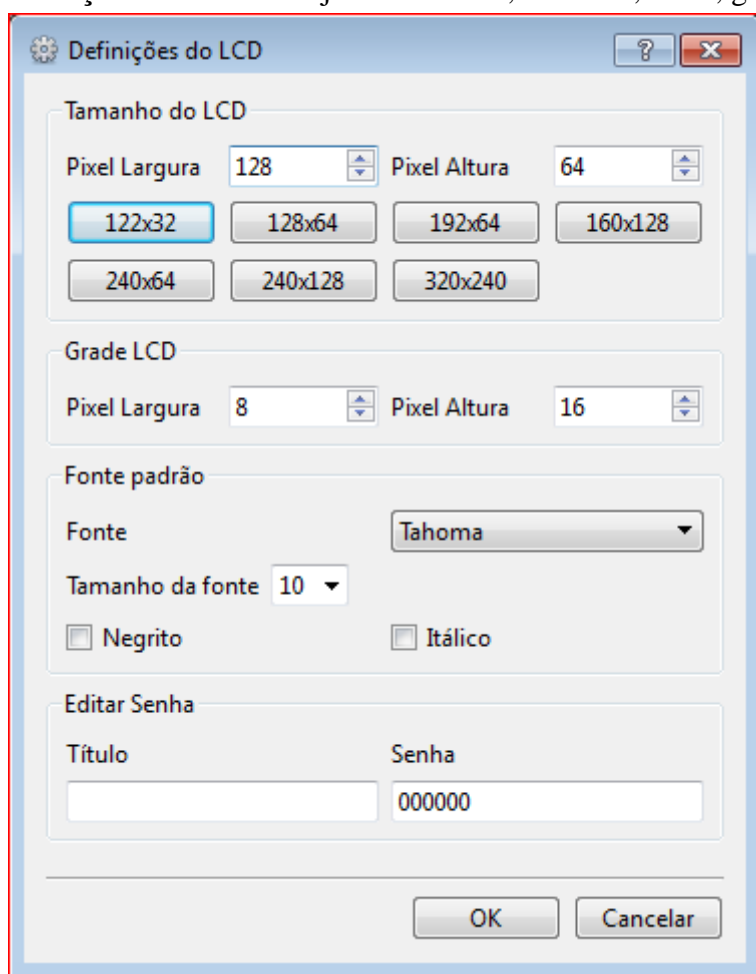
Exemplos :



Tipo de CLP destino: Selecione o modelo correto do CLP NeXo.



Definições do LCD : Ajustes do LCD, tamanho, fonte, grade e proteção.



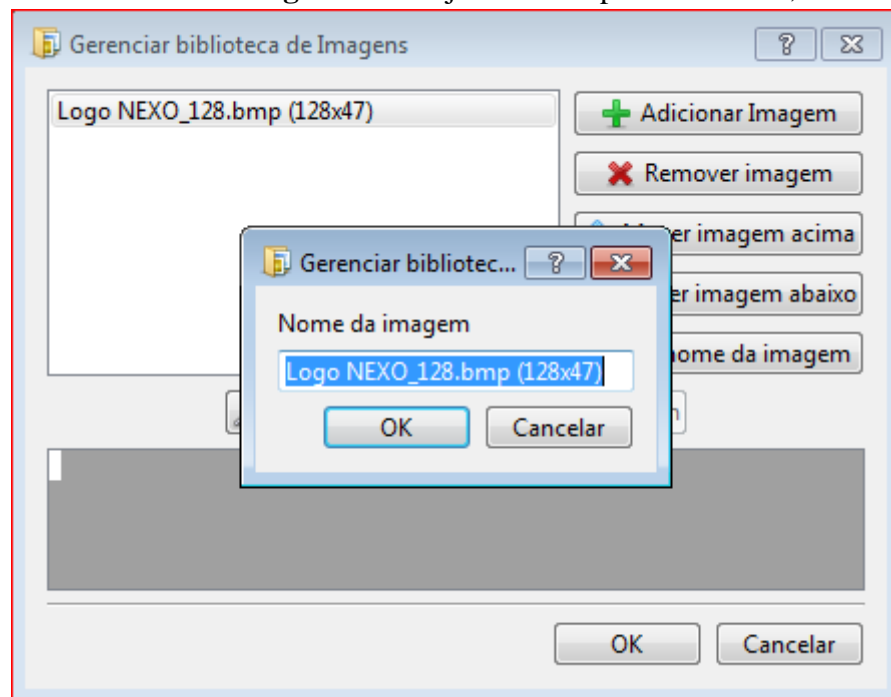
Tamanho do LCD : Ajuste o tamanho do LCD correspondente ao hardware.

Grade LCD : Aqui você pode ajustar o tamanho da Grade de Auxilio.

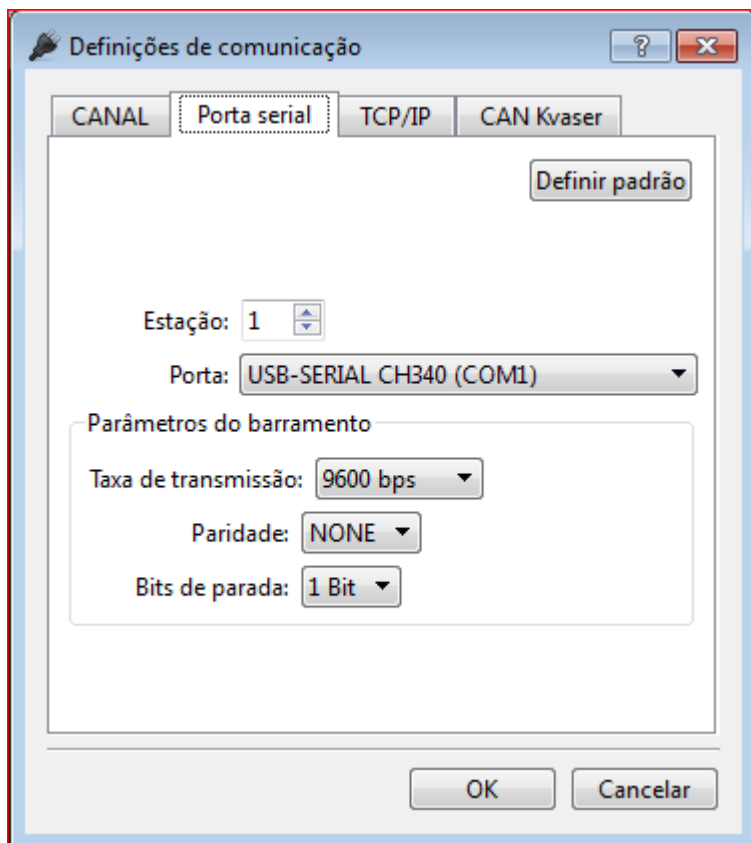
Fonte padrão: Define o tamanho e tipo de fonte padrão ao inserir um elemento.

Editar Senha: Você pode definir uma senha de proteção de 6 dígitos.

Gerenciador de imagens: Nessa janela você pode adicionar, renomear ou deletar suas imagens.



Definições de comunicação :



Atualmente só temos uma comunicação, Porta Serial.

Porta Serial:

Os valores padrão são:

Estação: 1.

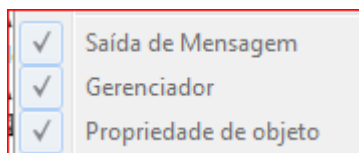
Porta : Selecione a porta onde o CLP está conectado.

Taxa de transmissão: 9600 bps.

Paridade : NONE.

Bits de parada : 1 Bit.

Janelas de interface:



Saída de Mensagem: Mostra informações de compilação e localização de erros. Essa janela não pode ser fechada e nem movida.

Saída de Mensagem

Compilado 4595 byte(s) [salvo na área de transferência]
Compilado 4595 byte(s) [salvo na área de transferência]
Compilado 4595 byte(s) [salvo na área de transferência]
Compilado 4595 byte(s) [salvo na área de transferência]
Compilado 4595 byte(s) [salvo na área de transferência]

Gerenciador:

Gerenciador			
Nome do objeto	X, Y	W, H	
Mostrar Página [1]			
Data [1]	66, 50	61,	13
imagem estático ...	0, -1	128,	47
Tempo [3]	1, 50	50,	13
Mostrar Página [2]			
Variável [1]	0, 0	42,	16
Texto estático [2]	121, 48	7,	16
Mostrar Página [3]			
Texto estático [1]	121, 48	7,	16
Variável [2]	0, 0	42,	16

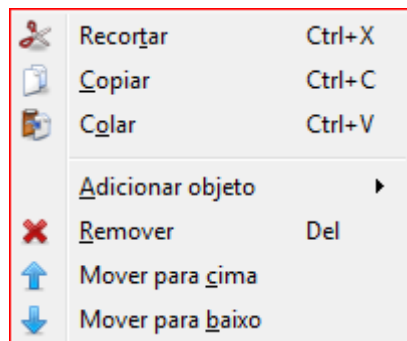
Para visualização e controle das paginas e elementos.

Propriedade dos objetos:

Propriedade de objeto						
Vazia	Página	Variável	Data	Hora	Lista de Textos	Lista de Imagens
Página			<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 4		
Grupo			<input type="checkbox"/> 1	<input type="checkbox"/> 5		
			<input type="checkbox"/> 2	<input type="checkbox"/> 6		
			<input type="checkbox"/> 3	<input type="checkbox"/> 7		

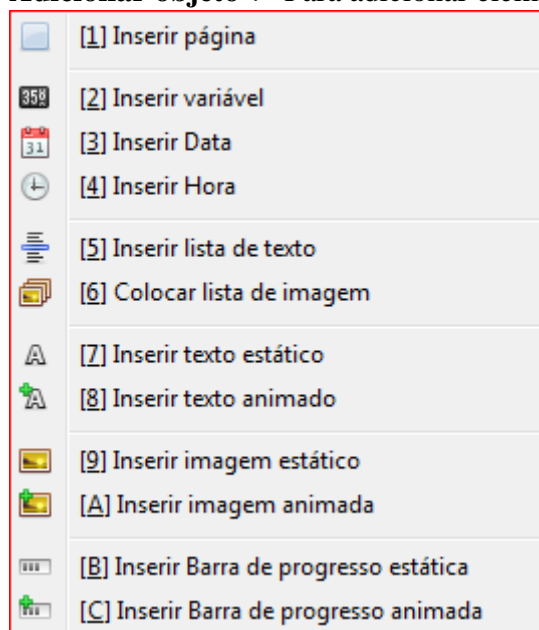
As propriedades de cada elemento inserido na página podem ser visualizadas e editadas aqui.

1.3 Editar



Recortar, copiar e colar : Você pode recortar, copiar e colar componentes no projeto.

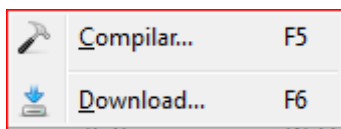
Adicionar objeto : Para adicionar elementos e páginas.



Remover : Para remover páginas e elementos.

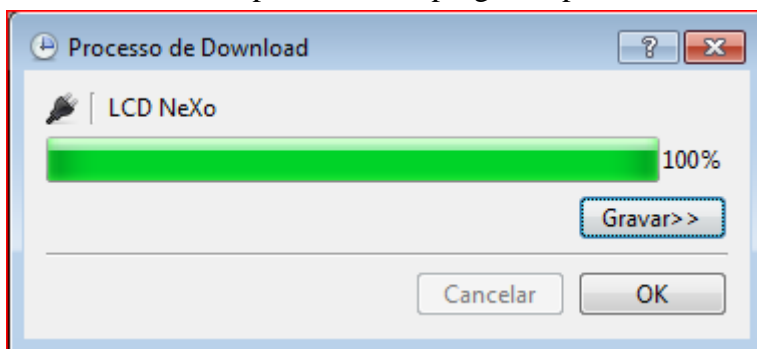
Mover para cima e para baixo: Para ajustar a posição das paginas e elementos no Gerenciador.

1.4 Ferramenta

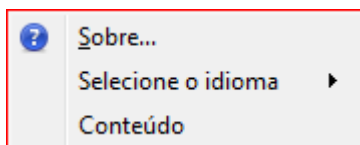


Compilar : Verifica erros e prepara o programa para ser transferido para o LCD.

Download : Compila e envia o programa para o LCD.



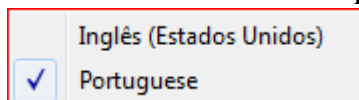
1.5 Ajuda



Sobre: Informação do software.



Selecione o idioma: Você pode mudar o idioma do programa entre Português e Inglês.



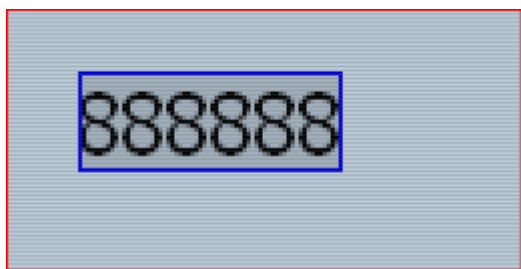
2. Barra de Ferramentas



Essa é a barra de ferramentas principal, todas as funções disponíveis para colocar na tela do LCD podem ser encontradas aqui

2.1 Variável





Clique em variável e ajuste as propriedades na janela da direita:

Propriedade de objeto

Vazia | Página | Variável | Data | Hora | Lista de Textos | Lista de Imagens

Básico

Coordenar X 18 | Coordenar Y 15

Mostrar | Inverter

Fonte

Fonte Tahoma

Tamanho da fonte 16

Negrito | Itálico

Variável

Ativar Edição | Editar Auto

Profundidade 1

Tipo de Valor INT

Largura da Fonte 6

Dígito 0

Faixa de valor Min -30000 | Max 30000

Endereço variável do CLP

Básico:

Coordenada: Coordenadas X e Y (vertical e horizontal) da posição do elemento na tela do LCD.

Mostrar : Mostrar ou ocultar o componente. (Fundo branco e texto preto)

Inverter : Inverter as cores do componente (fundo preto e texto branco)

Fonte : Ajustar propriedades da fonte, tamanho, tipo, etc.

Variável:

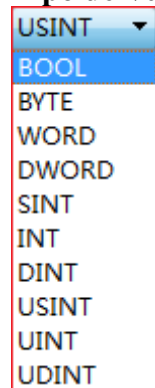
Ativar Edição : Habilita edição dessa variável pelo teclado do LCD.

Editar por senha : Habilita edição dessa variável pelo teclado por meio de uma senha que deverá ser inserida antes de inserir o valor da variável.

Editar Auto : Habilita edição com as teclas.

Profundidade: 1 indica que a edição será restrita na unidade. 2 indica que a edição será restrita na dezena, e assim por diante.

Tipo de Valor: Selecione o tipo de dado dessa variável.



A dropdown menu with a red border. The top item is 'USINT' with a small downward arrow. The second item, 'BOOL', is highlighted in blue. Below it are the following items: 'BYTE', 'WORD', 'DWORD', 'SINT', 'INT', 'DINT', 'USINT', 'UINT', and 'UDINT'.

Dígitos : Numero de dígitos máximo da variável .

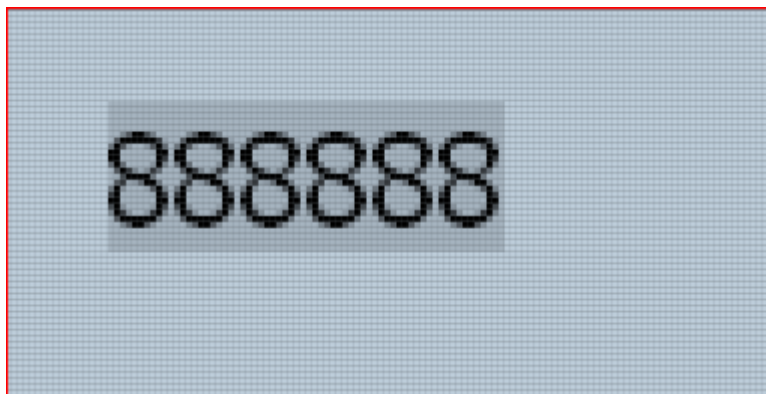
Decimal : Numero de casas decimais.

Faixa de valor : Intervalo dos valores (máximo e mínimo).

Endereço variável do CLP: Insira o endereço correspondente da variável no CLP.

Atenção : O tipo da variável deve ser consistente com a do CLP.

Exemplo :



Variável

Ativar Edição Editar Auto

Profundidade 1

Tipo de Valor INT

Largura da Fonte 6

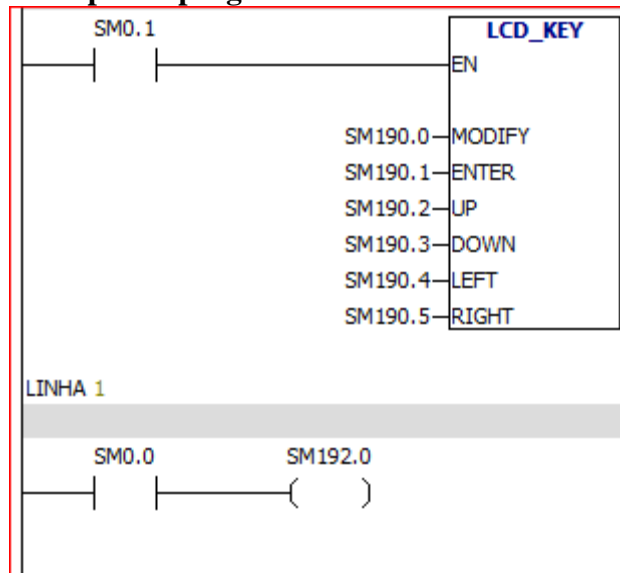
Dígito 0

Faixa de valor
Min -30000
Max 30000

Endereço variável do CLP

O intervalo de valores é -30000~30000. Isso limita a entrada de dados entre -30000 e 30000.

Exemplo de programa do CLP:



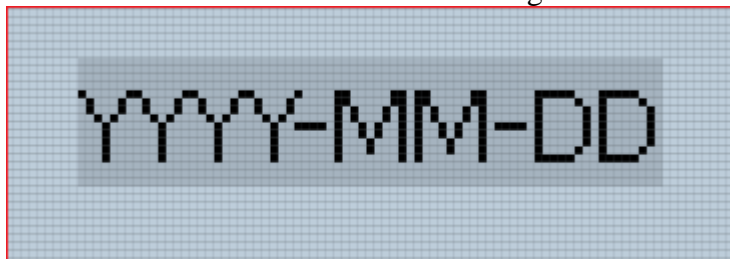
A LINHA 0 acima Mapeia os botoes do LCD.

A LINHA 1 faz com que o backlight do LCD fique sempre ligado.

2.2 Data

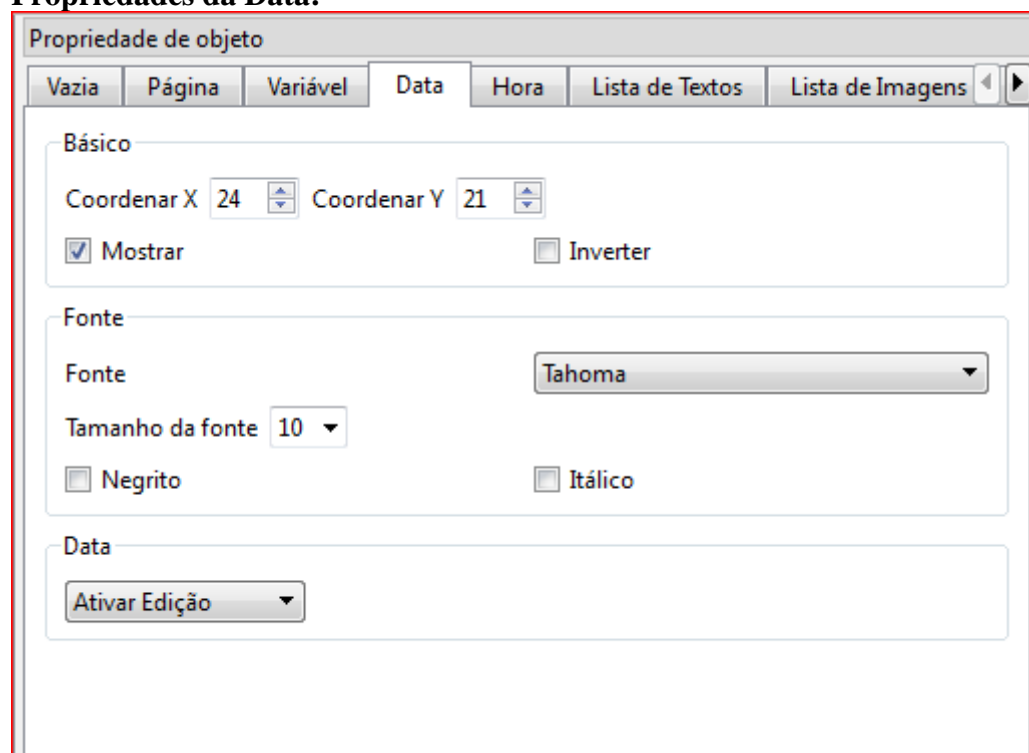


O elemento Data mostra na tela a data registrada no CLP:



A Data é mostrada automaticamente sem a necessidade de indicar uma variável.

Propriedades da Data:



Básico:

Coordenada : Posição X,Y do elemento.

Mostrar : Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Fonte:

Fonte: Escolher o tipo de fonte do elemento.

Tamanho da Fonte: Ajustar o tamanho da fonte do elemento.

Negrito e Itálico: Definir fonte como negrito e Itálico.

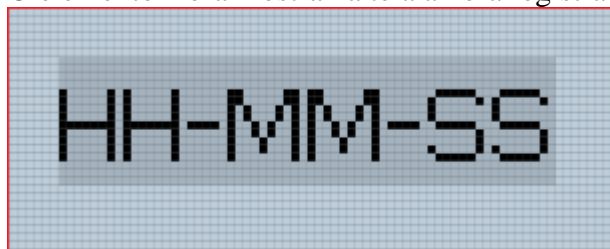
Data :

Ativar edição com ou sem senha, e desativar edição.

2.3 Hora



O elemento Hora mostra na tela a hora registrada no CLP:



Assim como a Data, o elemento Hora mostra a hora automaticamente sem a necessidade de indicar uma variável.

Propriedades da Hora:

Propriedade de objeto

Vazia | Página | Variável | Data | **Hora** | Lista de Textos | Lista de Imagens

Básico

Coordenar X 26 | Coordenar Y 20

Mostrar | Inverter

Fonte

Fonte Tahoma

Tamanho da fonte 10

Negrito | Itálico

Hora

Ativar Edição

Básico:

Coordenada : Posição X e Y do elemento.

Mostrar : Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Fonte:

Fonte: Escolher o tipo de fonte do elemento.

Tamanho da Fonte: Ajustar o tamanho da fonte do elemento.

Negrito e Itálico: Definir fonte como negrito e Itálico.

Hora:

Ativar edição com ou sem senha, e desativar edição.

2.4 Lista de Textos



Propriedades da Lista de Textos:

Propriedade de objeto

Vazia | Página | Variável | Data | Hora | Lista de Textos | Lista de Imagens

Básico

Coordenar X 0 Coordenar Y 0

Mostrar Inverter

Fonte

Fonte Tahoma

Tamanho da fonte 10

Negrito Itálico

Lista de Textos

Ativar Edição Editar loop

Endereço variável do CLP _____

Variável de bit _____

Lista de texto

- Laranja
- Banana
- Uva
- Melancia

+ Adicionar texto ✕ Remover texto

↑ Mover texto acima ↓ Mover texto abaixo

Básico:

Coordenada : Posição X,Y do elemento.

Mostrar : Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Fonte:

Fonte: Escolher o tipo de fonte do elemento.

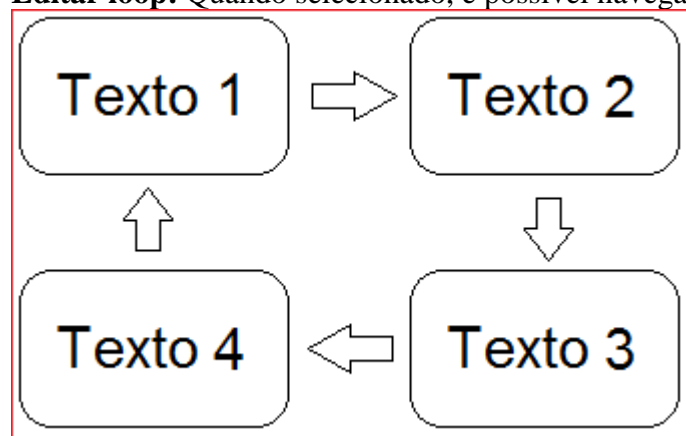
Tamanho da Fonte: Ajustar o tamanho da fonte do elemento.

Negrito e Itálico: Definir fonte como negrito e Itálico.

Lista de Textos:

Ativar edição com ou sem senha, e desativar edição..

Editar loop: Quando selecionado, é possível navegar pelos textos em um loop. Exemplo com 4 textos:

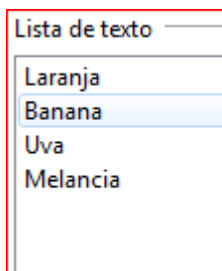


Endereços das variáveis no CLP:

Variável de Bit: Quando selecionado os texto são modificados por uma variável booliana. E quando não selecionado os textos são modificados por uma word, e o texto mostrado sera de acordo com o valor da variável.

Lista de texto: Onde você adicionara seus textos desejados.

Exemplos:



Selecionando 'Variável de Bit' e, usando como referencia a variável 'Q0.0' :

Quando 'Q0.0' é igual a 0 o texto mostrado será '*Laranja*'

Quando 'Q0.0' é igual a 1 o texto mostrado será '*Banana*'

Não selecionando 'Variável de Bit' e, usando como referencia a variável 'VW0' :

Quando 'VW0' é igual a 0 o texto mostrado será '*Laranja*'

Quando 'VW0' é igual a 1 o texto mostrado será '*Banana*'

Quando 'VW0' é igual a 2 o texto mostrado será '*Uva*'

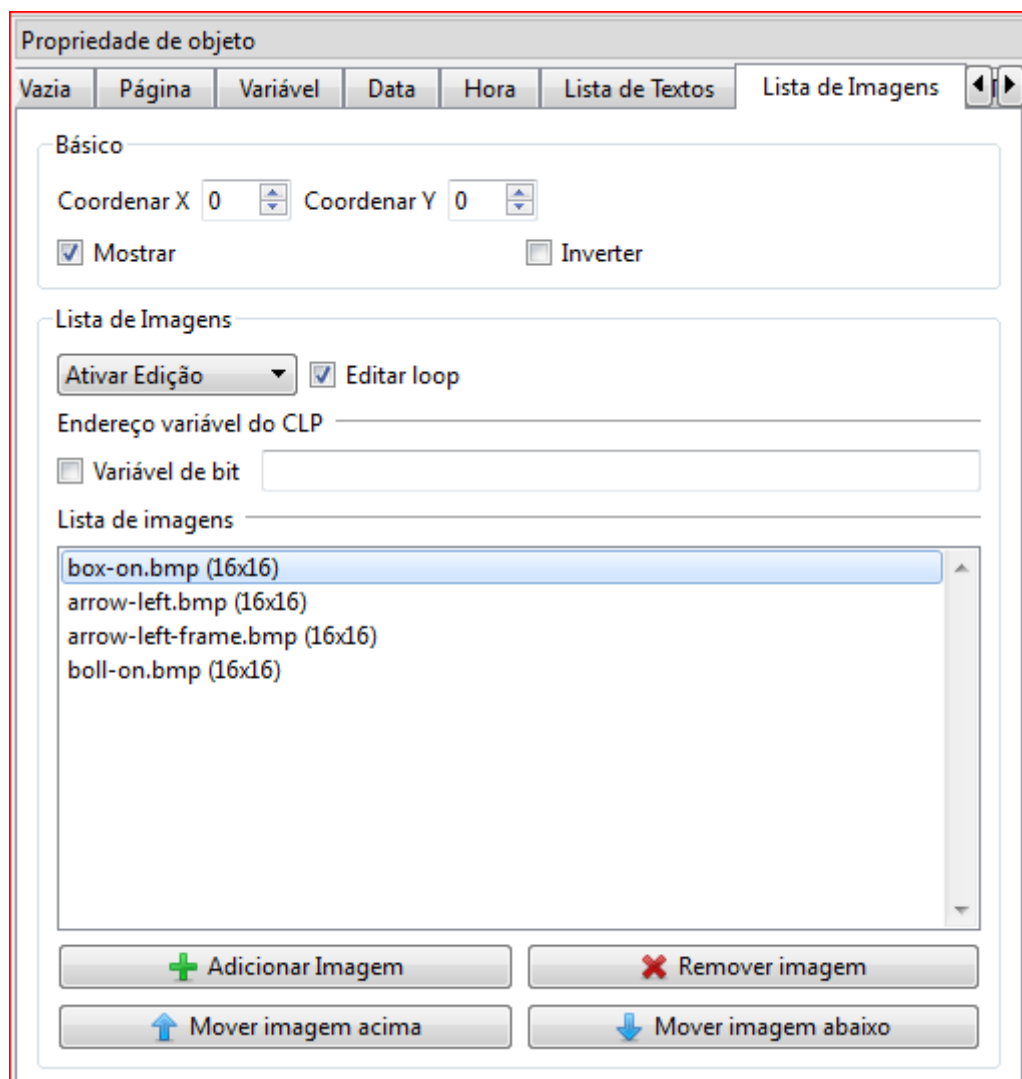
Quando 'VW0' é igual a 3 o texto mostrado será '*Melancia*'

Quando o valor de 'VW0' é maior que 3 não é mostrado nenhum texto.

2.5 Lista de Imagens



Lista de Imagens:



Básico:

Coordenada : Posição X,Y do elemento.

Mostrar : Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Fonte:

Fonte: Escolher o tipo de fonte do elemento.

Tamanho da Fonte: Ajustar o tamanho da fonte do elemento.

Negrito e Itálico: Definir fonte como negrito e Itálico.

Lista de Imagens:

Ativar edição com ou sem senha, e desativar edição..

Editar loop: Quando selecionado, é possível navegar pelas imagens em um loop. Assim como na 'Lista de Textos':

Endereços das variáveis no CLP:

Variável de Bit: Quando selecionado os imagens são modificados por uma variável booliana. E quando não selecionado, as imagens são modificadas por uma word, e a imagem mostrada será de acordo com o valor da variável.

Lista de imagens:

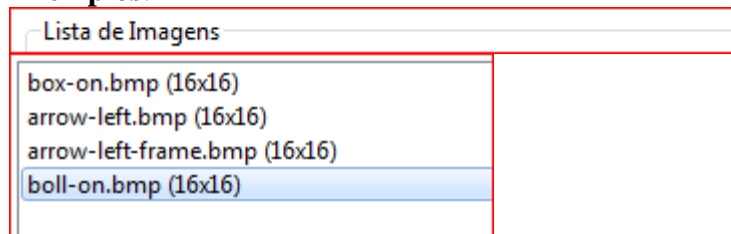
Adicionar Imagem: Clique para adicionar uma imagem à lista.

Remove Imagem: Remove a imagem selecionada da lista.

Mover Imagem Acima: Move a imagem uma posição para cima.

Mover Imagem Abaixo: Move a imagem uma posição abaixo.

Exemplos:



Bit, usando como referencia a variável 'Q0.0' :

Quando 'Q0.0' é igual a 0 a imagem mostrada será '*box-on.bmp(16x16)*'

Quando 'Q0.0' é igual a 1 a imagem mostrada será '*arrow-left.bmp (16x16)*'

Word, usando como referencia a variável 'VW0' :

Quando 'VW0' é igual a 0 a imagem mostrada será '*box-on.bmp(16x16)*'

Quando 'VW0' é igual a 1 a imagem mostrada será '*arrow-left.bmp (16x16)*'

Quando 'VW0' é igual a 2 a imagem mostrada será '*arrow-left-frame.bmp (16x16)*'

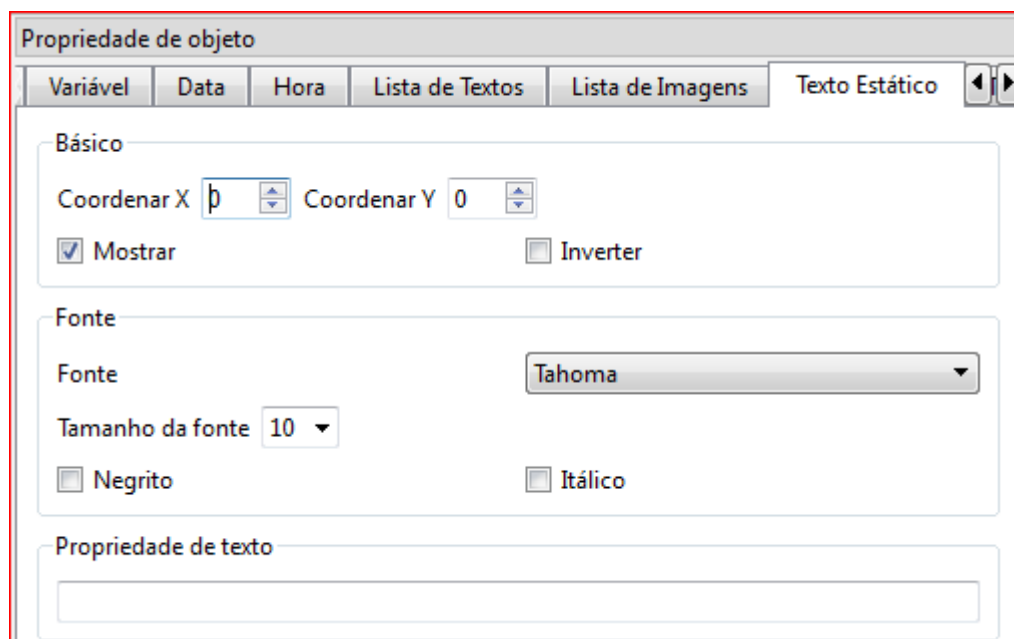
Quando 'VW0' é igual a 3 a imagem mostrada será '*boll-on.bmp (16x16)*'

Quando o valor de 'VW0' é maior que 3 não é mostrado nenhuma imagem.

2.6 Texto Estático



Texto Estático:



Básico:

Coordenada : Posição X,Y do elemento.

Mostrar : Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Fonte:

Fonte: Escolher o tipo de fonte do elemento.

Tamanho da Fonte: Ajustar o tamanho da fonte do elemento.

Negrito e Itálico: Definir fonte como negrito e Itálico.

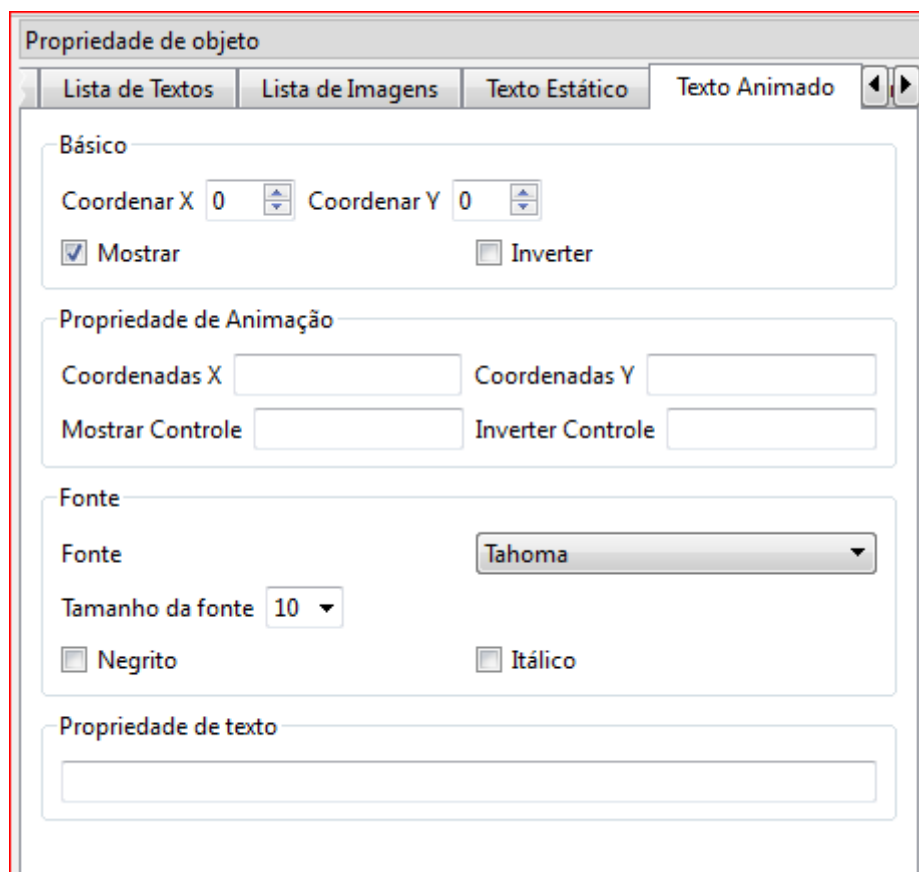
Propriedades de Texto:

Entre com o texto que deseja mostrar na tela.

2.7 Texto Animado



Texto Animado:



Básico:

Coordenada: Posição X,Y do elemento.

Mostrar: Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Propriedade de Animação:

Coordenadas X: Determine uma variável para fazer o movimento horizontal do elemento, quando o valor da variável diminui o elemento anda para esquerda, e quando aumenta anda para direita.

Coordenadas Y: Determine uma variável para fazer o movimento vertical do elemento, quando o valor da variável diminui o elemento anda para cima, e quando aumenta anda para baixo

Mostrar Controle: Mostra ou oculta o componente de acordo com o valor da variável bit.

Inverter Controle : Inverte as cores do componente de acordo com o valor da variável bit.

Fonte:

Fonte: Escolher o tipo de fonte do elemento.

Tamanho da Fonte: Ajustar o tamanho da fonte do elemento.

Negrito e Itálico: Definir fonte como negrito e Itálico.

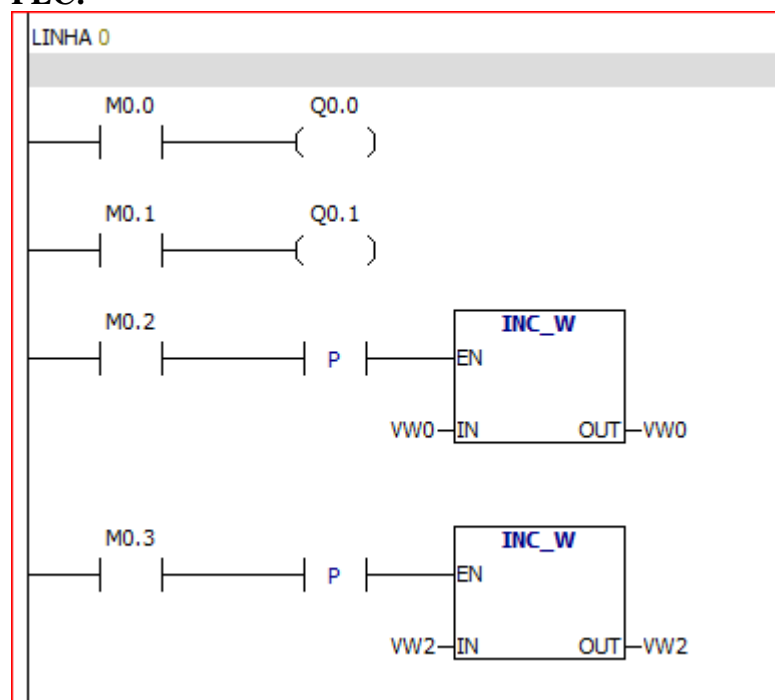
Propriedades de Texto:

Entre com o texto que deseja mostrar na tela.

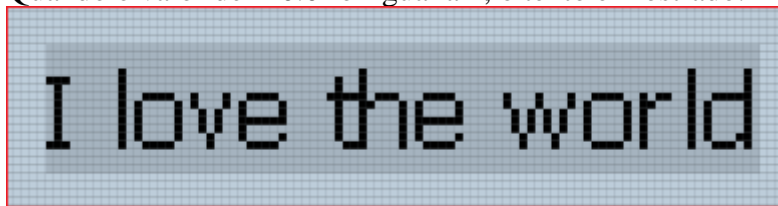
Exemplo:

The screenshot shows the 'Propriedade de objeto' dialog box with the 'Texto Animado' tab selected. The 'Básico' section includes 'Coordenar X' and 'Coordenar Y' set to 0, with 'Mostrar' checked and 'Inverter' unchecked. The 'Propriedade de Animação' section has 'Coordenadas X' set to 'vw0', 'Coordenadas Y' set to 'vw2', 'Mostrar Controle' set to 'q0.0', and 'Inverter Controle' set to 'q0.1'. The 'Fonte' section shows 'Fonte' as 'Tahoma', 'Tamanho da fonte' as '10', and both 'Negrito' and 'Itálico' unchecked. The 'Propriedade de texto' section contains a text input field with the text 'I love the world'.

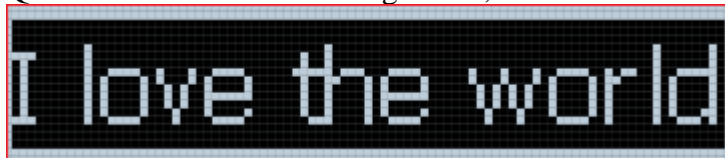
PLC:



Quando o valor de M0.0 for igual a 1, o texto é mostrado:



Quando o valor de M0.1 for igual a 1, o texto é mostrado com cores invertidas:



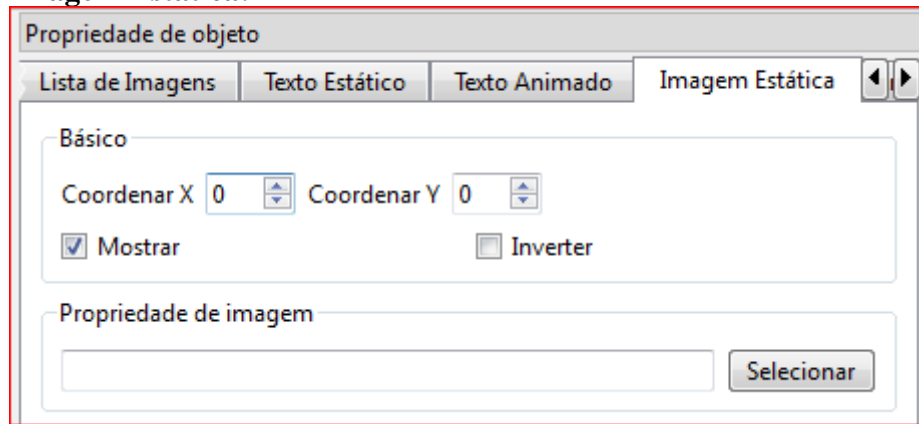
Cada vez que M0.2 é 1, o texto move um pixel para direita.

Cada vez que M0.3 é 1, o texto move um pixel para baixo.

2.8 Imagem Estática



Imagem Estática:



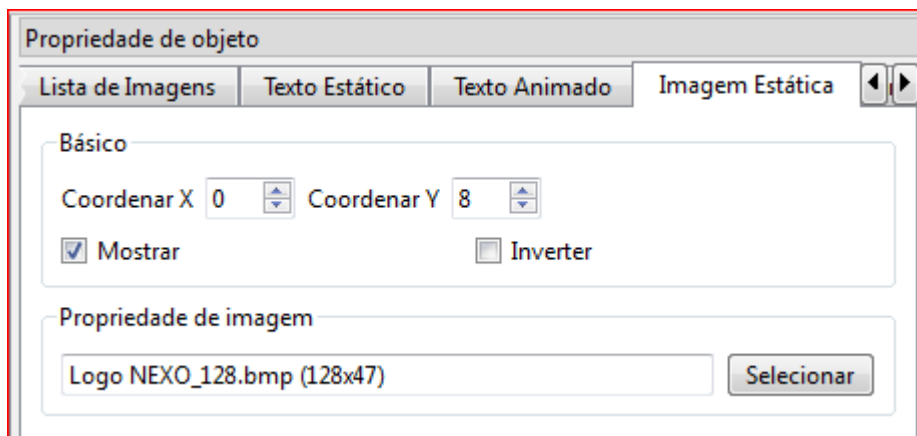
Básico:

Coordenada: Posição X,Y do elemento.

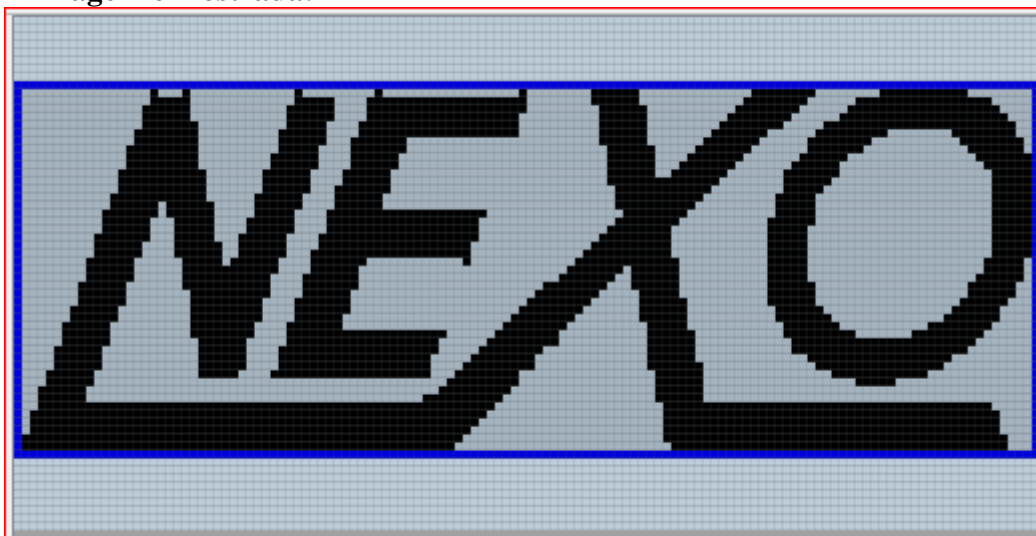
Mostrar: Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Exemplo:



A imagem é mostrada:



2.9 Imagem Animada



Propriedade da Imagem Animada:

Propriedade de objeto

Texto Estático | Texto Animado | Imagem Estática | Imagem Animada

Básico

Coordenar X 0 | Coordenar Y 0

Mostrar | Inverter

Propriedade de Animação

Coordenadas X | Coordenadas Y

Mostrar Controle | Inverter Controle

Propriedade de imagem

Selecionar

Básico:

Coordenada: Posição X,Y do elemento.

Mostrar: Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Propriedade de Animação:

Coordenadas X: Determine uma variável para fazer o movimento horizontal do elemento, quando o valor da variável diminui o elemento anda para esquerda, e quando aumenta anda para direita.

Coordenadas Y: Determine uma variável para fazer o movimento vertical do elemento, quando o valor da variável diminui o elemento anda para cima, e quando aumenta anda para baixo

Mostrar Controle: Mostra ou oculta o componente de acordo com o valor da variável bit.

Inverter Controle: Inverte as cores do componente de acordo com o valor da variável bit.

Propriedades de Imagem:

Selecione a Imagem para inserir na tela.

Exemplo:

Propriedade de objeto

Texto Estático | Texto Animado | Imagem Estática | Imagem Animada

Básico

Coordenar X 13 Coordenar Y 18

Mostrar Inverter

Propriedade de Animação

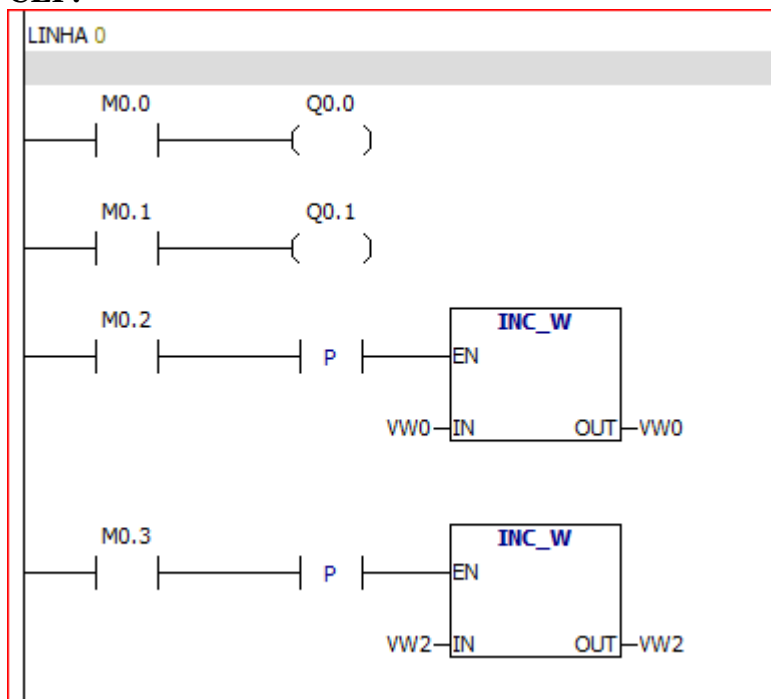
Coordenadas X vw0 Coordenadas Y vw2

Mostrar Controle Q0.0 Inverter Controle q0.1

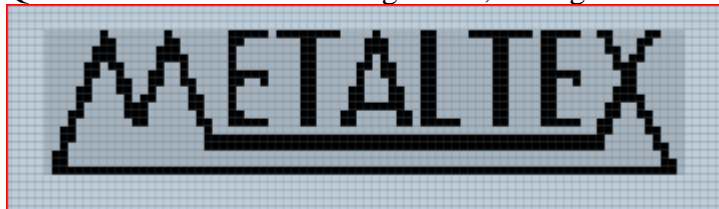
Propriedade de imagem

Logo MetaltexPB.bmp (80x18) Selecionar

CLP:



Quando o valor de M0.0 for igual a 1, a imagem é mostrada:



Quando o valor de M0.1 for igual a 1, a imagem é mostrada com cores invertidas:



Cada vez que M0.2 é 1, o texto move um pixel para direita.
Cada vez que M0.3 é 1, o texto move um pixel para baixo.

2.10 Barra de Progresso Estática



Propriedades da barra estática:

Propriedade de objeto

to Imagem Estática Imagem Animada Barra de Progresso Estática

Básico

Coordenar X 43 Coordenar Y 23

Mostrar Inverter

Propriedade de barra de progresso

Largura 32

Altura 8

Faixa de valor Min 0 Max 100

Endereço variável do CLP

Básico:

Coordenada: Posição X,Y do elemento.

Mostrar: Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

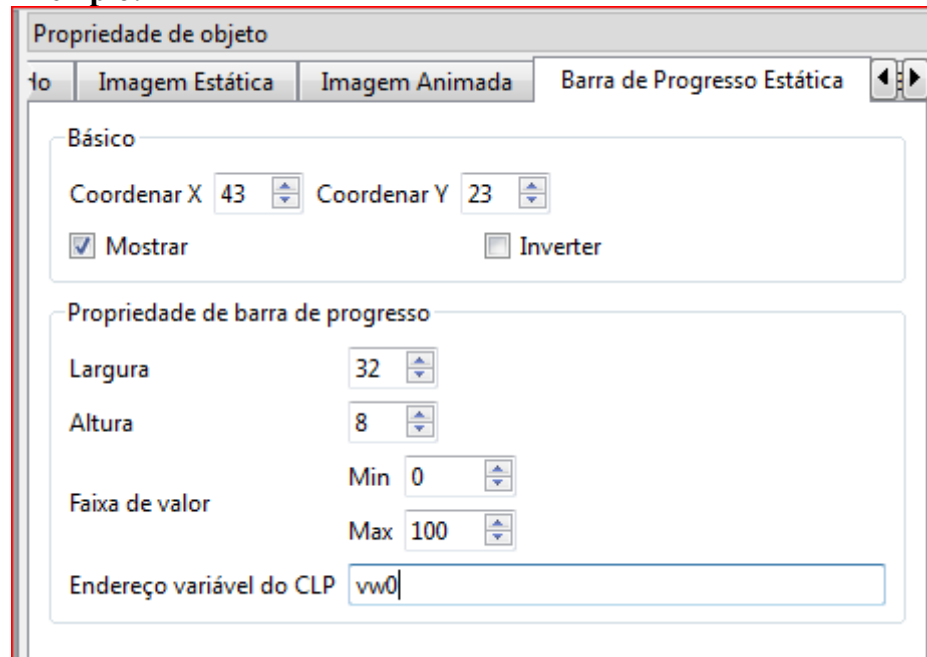
Propriedade da barra de progresso:

Largura e Altura: Para ajustar o tamanho da Barra de Progresso.

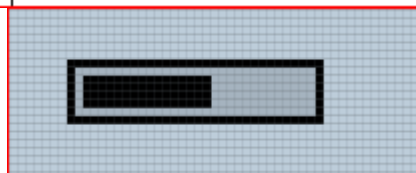
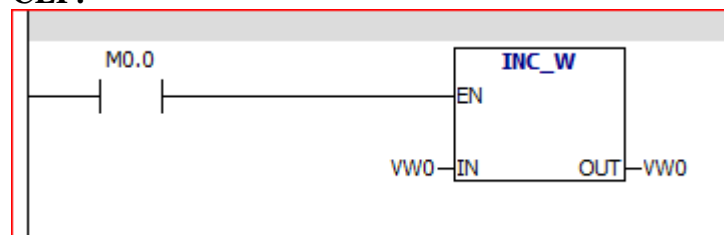
Faixa de Valor: Ajustar os valores máximo e mínimo da barra. Sendo de -32768 a 32767.

Endereço variável do CLP: Insira a variável desejada do CLP.

Exemplo:



CLP:



Quando o valor de VW0 for maior que 100, a barra mostra 100%. Quando o valor for menor que 0, a barra mostrará 0%.

2.11 Barra de Progresso Animada



Barra de Progresso animada:

Propriedade de objeto

em Animada Barra de Progresso Estática Barra de Progresso Animada

Básico

Coordenar X 35 Coordenar Y 18

Mostrar Inverter

Propriedade de Animação

Coordenadas X Coordenadas Y

Mostrar Controle Inverter Controle

Propriedade de barra de progresso

Largura 32

Altura 8

Faixa de valor Min 0 Max 100

Endereço variável do CLP

Básico:

Coordenada: Posição X,Y do elemento.

Mostrar: Mostrar ou ocultar o elemento.

Inverter: Inverte a cor do elemento.

Propriedade de Animação:

Coordenadas X: Determine uma variável para fazer o movimento horizontal do elemento, quando o valor da variável diminui o elemento anda para esquerda, e quando aumenta anda para direita.

Coordenadas Y: Determine uma variável para fazer o movimento vertical do elemento, quando o valor da variável diminui o elemento anda para cima, e quando aumenta anda para baixo

Mostrar Controle: Mostra ou oculta o componente de acordo com o valor da variável bit.

Inverter Controle: Inverte as cores do componente de acordo com o valor da variável bit.

Propriedade da barra de progresso:

Largura e Altura: Para ajustar o tamanho da Barra de Progresso.

Faixa de Valor: Ajustar os valores máximo e mínimo da barra. Sendo de -32768 a 32767.

Endereço variável do CLP: Insira a variável desejada do CLP.

Exemplo:

Propriedade de objeto

em Animada | Barra de Progresso Estática | Barra de Progresso Animada

Básico

Coordenar X 35 Coordenar Y 18

Mostrar Inverter

Propriedade de Animação

Coordenadas X vw0 Coordenadas Y vw2

Mostrar Controle q0.0 Inverter Controle q0.1

Propriedade de barra de progresso

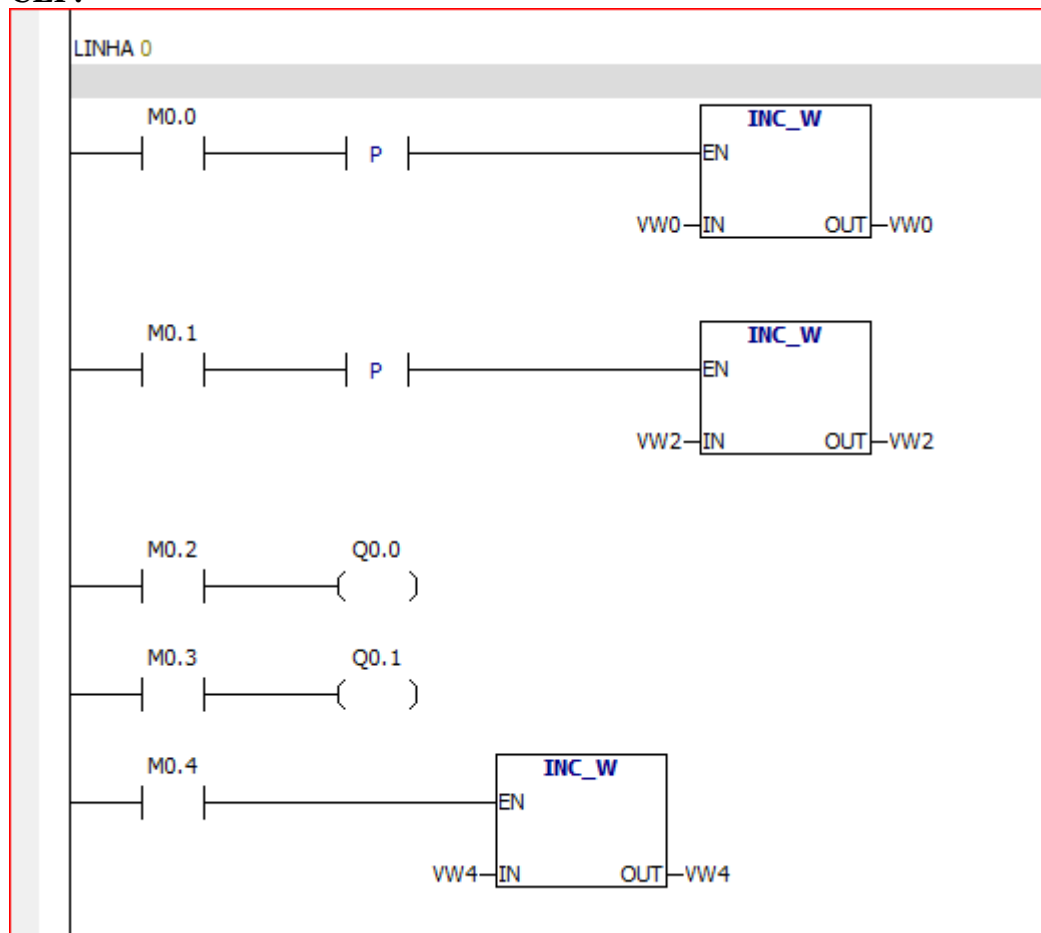
Largura 32

Altura 8

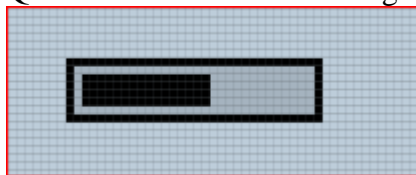
Faixa de valor Min 0 Max 100

Endereço variável do CLP vw4

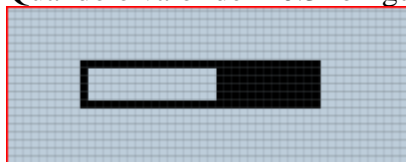
CLP:



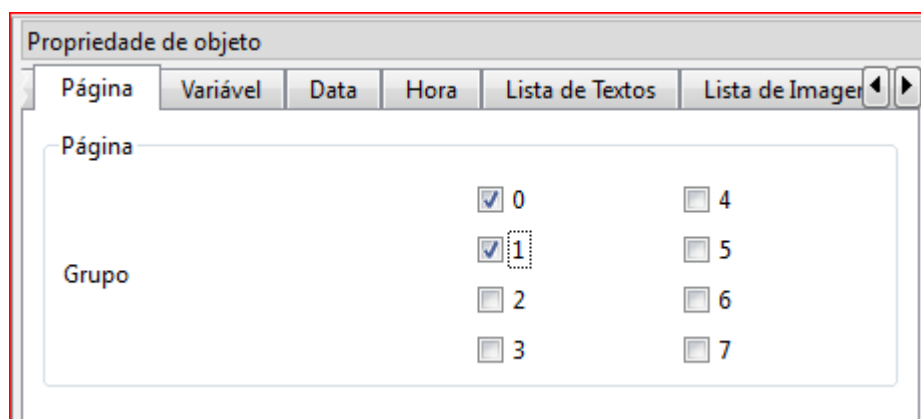
Cada vez que M0.0 é 1, a barra move um pixel para direita.
Cada vez que M0.1 é 1, a barra move um pixel para baixo.
Quando o valor de M0.2 for igual a 1, a barra é mostrada:



Quando o valor de M0.3 for igual a 1, a barra é mostrada com cores invertidas:



3. Propriedades da Página



Existem 8 grupos de paginas. Você pode selecionar a qual grupo pertence cada pagina, e especificar no CLP através da função LCD-PAGE, qual grupo estará disponível. As paginas dos outros grupos não estarão disponíveis.

Exemplo:

Temos 5 paginas:

Gerenciador		
Nome do objeto	X, Y	W, H
Mostrar Página [1]		
Barra de progress...	37, 25	50, 15
Mostrar Página [2]		
Variável [1]	18, 15	66, 25
Mostrar Página [3]		
Data [1]	24, 21	73, 16
Mostrar Página [4]		
Tempo [1]	26, 20	62, 16
Mostrar Página [5]		

Página 1: Vinculada ao Grupo 0 e Grupo 1.

Pagina 2: Vinculada ao Grupo 0.

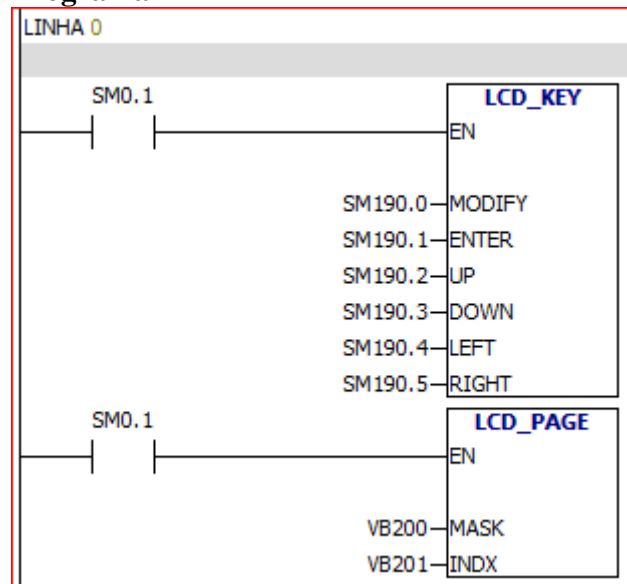
Pagina 3: Vinculada ao Grupo 0.

Pagina 4: Vinculada ao Grupo 1.

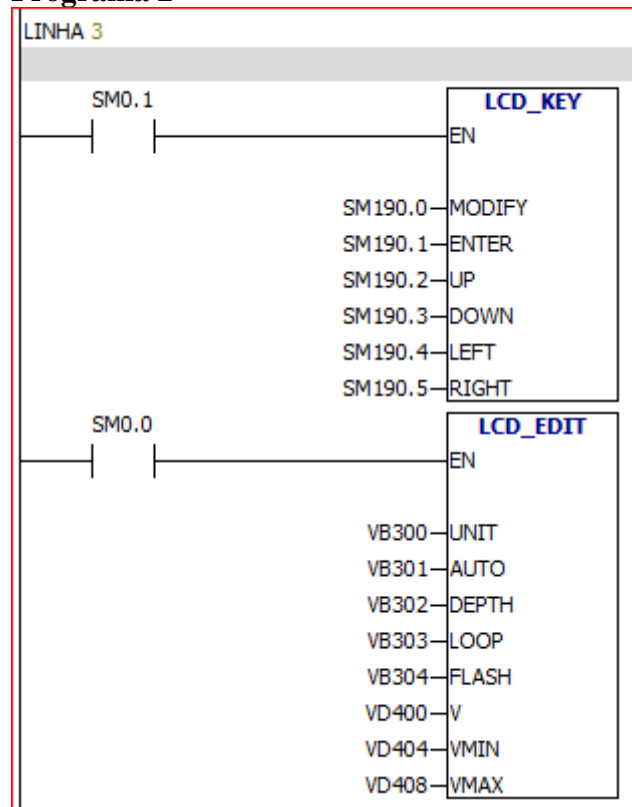
Pagina 5: Vinculada ao Grupo 2.

PLC :

Programa 1



Programa 2



Analizando:

No programa 1 A função de assimilação dos botoes é chamada, e também a função LCD-PAGE.

No programa 2 A função de assimilação dos botoes é chamada, e também a função LCD-EDIT.

Coloque o CLP em RUN e o valor de VB200 e VB201 como segue:

VB200	SINT	1
VB201	SINT	0

Com VB200 igual a 1, todas as paginas marcadas como grupo 0 estarão disponíveis: Paginas 1, 2 e 3.

Com VB200 igual a 2, todas as paginas marcadas como grupo 1 estarão disponíveis: Paginas 1 e 4.

Com VB200 igual a 3, todas as paginas marcadas como grupo 0 e grupo 1 estarão disponíveis: Paginas 1, 2, 3 e 4. Porque?

Assumindo que a VB200 é interpretada Bit a Bit, temos:

Grupo ⇒	0	1	2	3	4	5	6	7
Valor ⇒	1	2	4	8	16	32	64	128

Quando colocamos 3 (1+2) estamos selecionando os grupos 0 e 1.

Se colocarmos 7 (1+2+4) estaremos selecionando os grupos 0, 1 e 2.

A variável VB201 vai indicar e controlar o numero da pagina exibida, estando disponíveis somente as paginas dos grupos selecionados na VB200.

4. Variáveis do CLP

Variáveis dos Botoes no CLP:

ESC	>	SM190.0
OK	>	SM190.1
SETA A CIMA	>	SM190.2
SETA A BAIXO	>	SM190.3
ESQUERDA	>	SM190.4
DIREITA	>	SM190.5

Variável de controle da Iluminação da tela.

SM192.0 igual a 1, LCD aceso.

SM192.0 igual a 0, LCD apagado.

É preciso habilitar os botoes do LCD usando as funções disponíveis no CLP.

ESC é usado para editar as variáveis.

OK é usado para confirmar o valor editado e sair da edição.

SETAS A CIMA e A BAIXO são usadas para trocar as paginas e edição dos valores nas variáveis. Pressione ESC até chegar na variável que deseja editar, use as setas para alterar o valor e pressione OK para confirmar. Quando estiver editando alguma variável, é possível navegar entre as variáveis com as SETAS DIREITA E ESQUERDA, e também navegar entre os dígitos da variável caso a Profundidade seja maior que 1 (Veja tópico 2.1)