



CONTROLADORES PROGRAMÁVEIS INFINIUM

MANUAL DE PROGRAMAÇÃO

Versão 1.00
Abril de 2014

Infinium Automação Industrial Ltda
site: www.infiniumautomacao.com.br
email: contato@infiniumautomacao.com.br

1. APRESENTAÇÃO

O software SION é a ferramenta de configuração e programação de Controladores Programáveis da INFINIUM, baseado na norma IEC 61131-3, utiliza linguagem Ladder e permite a programação, monitoramento de processos, configuração de módulos e geração de históricos e gráficos.

Voltado à programação orientada a objetos, o software utiliza o conceito de programação simbólica, e possibilita criar sub rotinas para reutilização em diversos outros programas, otimizando o tempo de desenvolvimento de novas aplicações. A configuração do hardware do CP é intuitiva e de fácil utilização.

Características de Destaque

- **Software Gratuito**
- **Declaração de variáveis utilizando tabela de edição**
- **Diagrama ladder IEC-61131-3**
- **Diversos blocos funcionais garantindo maior flexibilidade na programação**
- **Várias opções de monitoração do programa aplicativo**

2. APLICAÇÕES DOS CONTROLADORES PROGRAMÁVEIS

Os controladores programáveis são equipamentos extremamente versáteis e possuem aplicações nos mais diversos ramos de atividades:

Transporte <ul style="list-style-type: none">- Sistemas de esteiras transportadoras- Plataformas de elevação- Elevadores- Controle de silos	Controle de máquinas <ul style="list-style-type: none">- Comando de motores, bombas e válvulas- Compressores de ar- Sistemas de exaustão e filtragem- Estações de tratamento de água- Serras e plainas
Gestão de casas e edifícios <ul style="list-style-type: none">- Controle de iluminação (interior e exterior)- Automação de portas e portões- Controle de persianas e toldos- Controle de sistemas de <i>sprinklers</i> e fornecimento de água	Soluções especiais <ul style="list-style-type: none">- Sistemas fotovoltaicos- Uso em navios- Uso sob condições ambientais severas- Controle de painéis de publicidade- Controle da sinalização de trânsito- Controle de irrigação
Aquecimento, Ventilação, Ar Condicionado <ul style="list-style-type: none">- Gerenciamento de energia- Controle de caldeiras- Sistemas de refrigeração- Sistemas de ventilação- Sistemas de ar condicionado.	Sistemas de monitoramento <ul style="list-style-type: none">- Controle de acesso- Monitoramento de veículos em estacionamentos- Sistemas de alarme

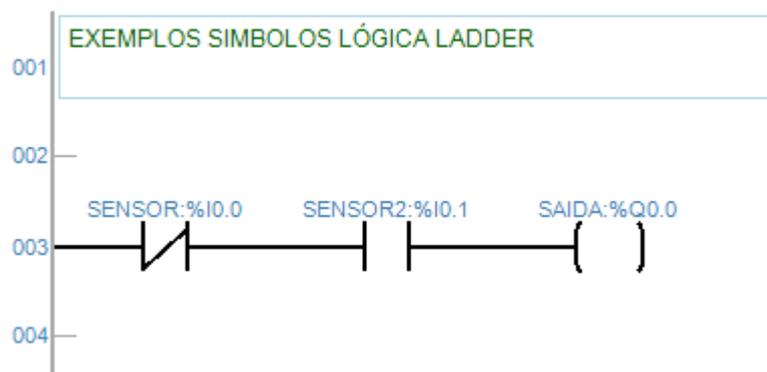
3. PROGRAMAÇÃO

O diagrama *ladder* utiliza lógica de relé, com contatos e bobinas, e por isso é a linguagem de programação de CLP mais simples de ser assimilada para quem já tenha conhecimento de circuitos de comando elétrico.

Compõe-se de vários circuitos dispostos horizontalmente, com a bobina na extremidade direita, alimentada por duas barras verticais laterais. Por esse formato é que recebe o nome de *ladder* que significa escada em inglês.

Cada uma das linhas horizontais é uma sentença lógica onde os contatos são as entradas das sentenças, as bobinas são as saídas e a associação dos contatos é a lógica.

São os seguintes os símbolos:



No *ladder* cada operando (nome genérico dos contatos e bobinas no *ladder*) é identificado com um endereço da memória à qual se associa no CLP. Esse endereço aparece no *ladder* com um nome simbólico, para facilitar a programação.

3.1 OPERANDOS E ENDEREÇOS

3.1.1 Operandos %I – Entradas (input)

São operandos usados para referenciar pontos de entradas digitais. Sua quantidade é determinada pelo número de módulos de E/S que estão dispostos nos módulos que compõem o sistema.

Os operandos %I são utilizados em instruções binárias (contatos, bobinas). Ocupam um byte de memória (8 bits), armazenando os valores dos pontos diretamente em cada bit.

%I0 – entrada digital 1

3.1.2 Operandos %Q – Saídas (output)

São operandos usados para referenciar pontos de saída digital. Sua quantidade é determinada pelo número de módulos de E/S que estão dispostos nos módulos que compõem o sistema. Os operandos %Q são utilizados em instruções binárias (contatos, bobinas). Ocupam um byte de memória (8 bits), armazenando os valores dos pontos diretamente em cada bit.

%Q2 – saída digital 3

3.1.3 Operandos %M – Memórias bits

As memórias de bits são operandos usados para armazenamento e manipulação de valores binários intermediários no processamento do programa aplicativo. Sua quantidade nos controladores é fixa. Operandos %M são utilizados em instruções binárias (contatos, bobinas). Armazenando valores diretamente em cada bit.

%M1 - memória 1 (tamanho 1bit)

3.1.4 Operandos %MB – Memórias de 8 bits (byte)

Os operandos %MB são usados para processamento numérico e armazenando de valores, na faixa de 0 a 255 decimal.

%MB1 – memória 1 (tamanho 8bits)

3.1.5 Operandos %MW – Memórias de 16 bits (word)

Os operandos %MW são usados para processamento numérico e armazenando de valores, na faixa de 0 a 65.535 decimal.

%MW2 – memória 2 (tamanho 16bits)

3.1.6 Operandos %MD – Memórias de 32 bits (dword)

Os operandos %MW são usados para processamento numérico e armazenando de valores, na faixa de 0 a 4.294.967.296 decimal.

%MD3 – memória 3 (tamanho 32bits)

3.1.7 Operandos Timers

%T – Acumuladores Timer

Os operandos %Tx.V são usados para armazenamento do valor do acumulador dos timer's usados no programa.

%T2 – valor do acumulador do timer 2

3.1.8 Operandos Contadores

%C – Acumuladores Contadores

Os operandos %C são usados para armazenamento do valor do acumulador dos contadores usados no programa.

%C10 – valor do acumulador do contador 10

3.1.9 Operandos %IW – Entradas analógicas (Word de entrada)

Os operandos %IW são usados para armazenamento do valor das entradas analógicas.

%IW6 – entrada analógica 6

A faixa de valores vai de 0 a 4095, isto para valor bruto, ou seja, sem a linearização de escala, o que pode ser feito em “fundo de escala” na aba configurações do projeto.

3.1.10 Operandos %QW – Saídas analógicas (Word de saída)

Os operandos %QW são usados para manipular o valor das saídas analógicas.

%QW2 – saída analógica 2

A faixa de valores vai de 0 a 4095. O tipo de saída pode ser 0-20mA ou 4-20mA configurada na aba “configurações do projeto”.

3.1.11 Operandos %Sx – Bits do Sistema

Os operandos %Sx são usados como operandos auxiliares e cada um realiza uma função específica. Como visto na seqüência.

%S0 - Sempre ligado

%S1 - Sempre desligado

%S2 - Ligado na primeira varredura

%S3 - Desligado na primeira varredura

%S4 - Clock de 0,5 Segundos

%S5 - Clock de 30 Segundos

%S6 - CLP RUN / STOP (apenas leitura)

%S7 - Tecla < + > da IHM Easy View

%S8 - Tecla < - > da IHM Easy View

%S9 - Tecla < ENTER > da IHM Easy View

%S10 - Tecla < SHIFT > da IHM Easy View

%S11 - Link de RF (0: sem link, 1:link ok, se servidor sempre 1), apenas para Conn-FLEX.

3.1.12 Operandos %Bx – Bytes do Sistema

Os operandos %Bx são usados como operandos auxiliares e cada um realiza uma função específica. Como visto na seqüência.

%B0 - Segundos

%B1 - Minutos

%B2 - Horas

%B3 - Dia da Semana

%B4 - Dia

%B5 - Mês

%B6 - Ano

%B7 - Tempo de Scan (décimos de segundo)

%B8 - Status de Comunicação COM1

%B9 - Status de Comunicação COM2

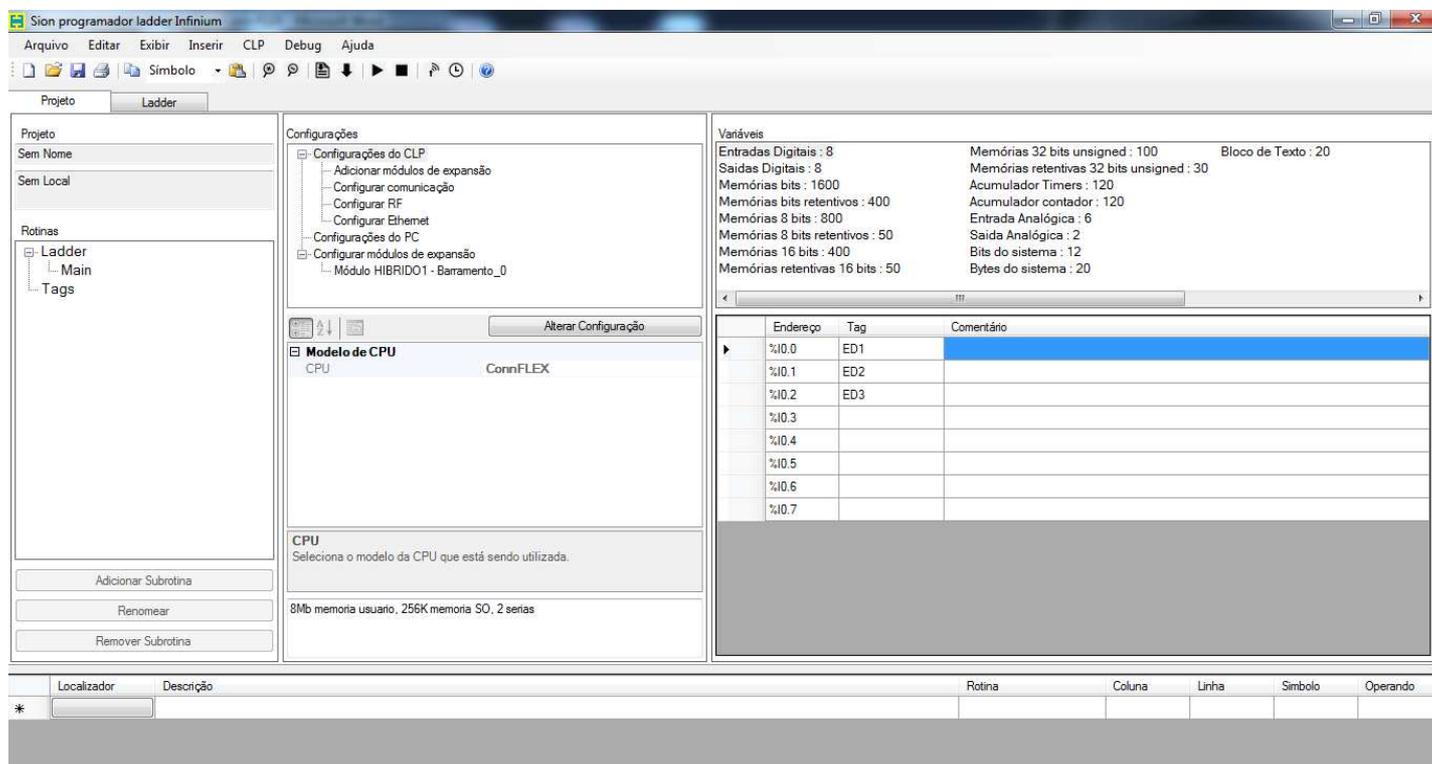
%B10 - Status de Comunicação COM3

3.1.13 Operandos %Tx – Texto

Os operandos %Tx são operandos usados quando o controlador possuir IHM integrada, neles são armazenados as mensagens de texto que serão apresentadas na IHM do equipamento.

3.2 CONFIGURAÇÃO E PROGRAMAÇÃO

Através do *software* de programação SION é feita toda a configuração e programação do controlador programável. SION é compatível com *Windows XP, VISTA, e Windows 7.*



Para o correto funcionamento do software SION é necessário instalar antes o Microsoft Framework 3.0 ou superior.

Dentro da aba PROJETO encontram-se as janelas:

PROJETO: são exibidas informações do projeto, como nome, local onde está salvo, ainda possui as funções de adicionar/excluir rotinas e listar as rotinas pertencentes ao projeto.

CONFIGURAÇÕES: Nesta opção são configurados o tipo de CPU, configura os módulos de hardware pertencentes ao CLP, configura as interfaces de comunicação, configuração da porta COM usada no PC para programar o CLP.

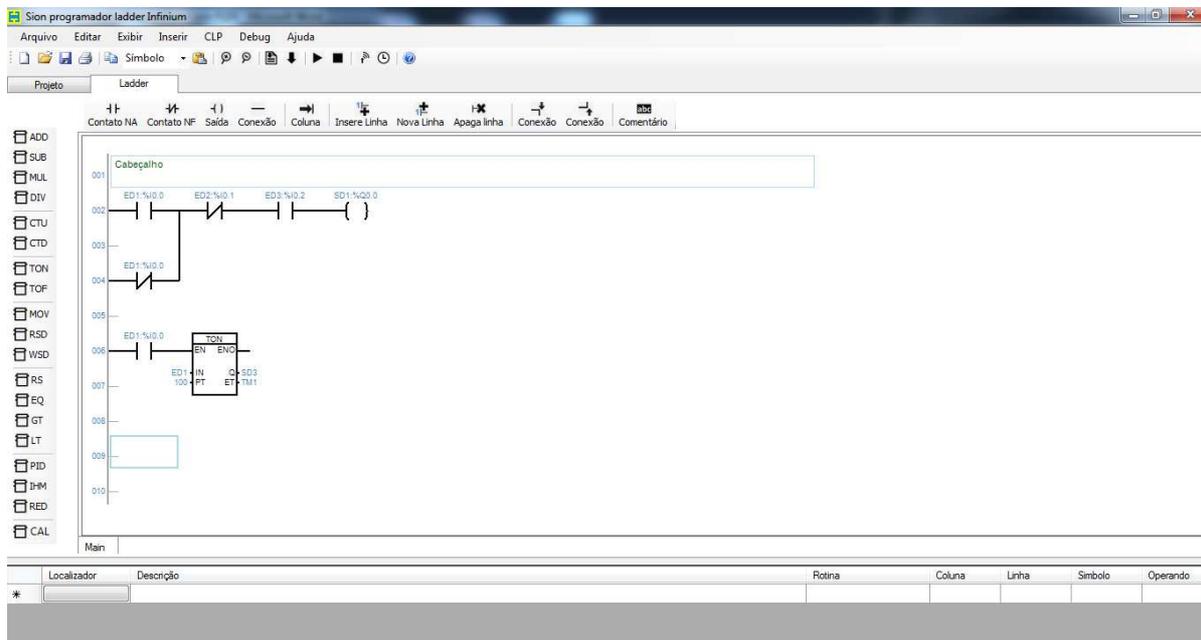
VARIÁVEIS: Nesta área são listadas os operandos / variáveis do CLP, nela pode-se associar nomes aos operandos para facilitar a programação e identificação no programa ladder.

Botões da barra de ferramentas:

-  Zoom mais e zoom menos para diagrama *ladder*
-  Localiza variável no programa *ladder*.
-  Compila o programa *ladder* corrente e identifica possíveis erros mostrando no inferior da tela.
-  Grava programa *ladder* para o CLP.
-  Passa o CLP para modo de execução (RUN)
-  Passa o CLP para bloqueado, parando a execução do programa aplicativo (STOP)
-  Configura interface de RF, caso o CLP seja equipado com transceptor de rádio (*wireless*), após a transferência do programa deve-se clicar neste botão para que o CLP faça as alterações nos parâmetros referentes à parte de RF.
-  Ajusta o Relógio do CLP de acordo com o relógio do computador que está rodando o *software* Sion.
-  Monitora variáveis e o programa aplicativo do usuário.
-  Gera gráfico e histórico das variáveis selecionadas do programa aplicativo.
-  Abre os manuais dos CLP's e de Programação.

3.3 LADDER

O diagrama *ladder* segue a norma IEC 61131-3. Seus elementos são descritos na seqüência que segue.



3.3.1 Contato Aberto

Contato aberto aceita todos os operandos do tipo bit. Acima do símbolo gráfico é mostrado a TAG e endereço.



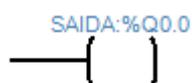
3.3.2 Contato Fechado

Contato fechado aceita todos os operandos do tipo bit. Acima do símbolo gráfico é mostrado a TAG e endereço.



3.3.3 Bobinas

Aceitam os operandos do tipo bit. Acima do símbolo gráfico é mostrado a TAG e endereço.



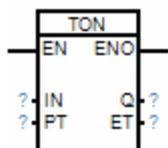
3.3.4 Temporizador na Energização (TON)

Bloco funcional destinado a realizar temporizações quando habilitado. Com as entradas EN e IN habilitadas, inicia-se a contagem de tempo na saída ET (Base de tempo décimos de segundo) até o valor de preset (PT). Durante essa contagem, a saída Q permanece desligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

Se EN for desligado antes do fim da temporização, a temporização entra em PAUSE, voltando ao normal quando EN for novamente habilitado.

O RESET de temporização acontece quando a entrada IN for desligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado. O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q ligada.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR, %S	Habilita execução da instrução. Pausa temporização ao desabilitar antes do fim da contagem de tempo.
	IN	%I, %Q, %M, %MR, %S	Iniciar temporização.
	PT	Constante	Preset do Temporizador
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	Q	%Q, %M, %MR	Habilitado no fim da temporização
	ET	%Tx.V	Efetivo do temporizador

3.3.5 Temporizador na Desenergização (TOF)

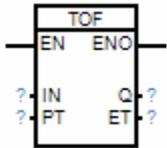
Com as entradas EN e IN habilitadas, na transição de TRUE para FALSE da entrada IN, inicia-se a contagem de tempo na saída ET (Base de tempo décimos de segundo) até o valor de preset (PT). Durante essa contagem, a saída Q permanece ligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será desligada e permanecerá assim até que IN seja ligado. Quando a entrada IN é ligada, a saída Q é ligada novamente e o efetivo ET zerado (ET=0).

Se IN for desligado antes do fim da temporização, a temporização é PAUSADA, voltando ao normal quando IN for novamente habilitado.

O RESET de temporização acontece quando a entrada EN for desligada.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q desligada.



E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR, %S	Habilita execução da instrução. Pausa temporização ao desabilitar antes do fim da contagem de tempo.
	IN	%I, %Q, %M, %MR, %S	Iniciar temporização
	PT	Constante	Preset do Temporizador
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	Q	%Q, %M, %MR	Desabilitado no fim da temporização
	ET	%Tx.V	Efetivo do temporizador

3.3.6 Contador Crescente (CTU)

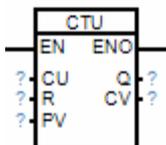
A cada pulso em CU, a saída CV será incrementada.

Enquanto o valor da saída CV for menor que o valor de preset (PV), a saída Q permanecerá desligada. Quando a saída CV for igual ao valor de preset (PV), a saída Q será ligada.

Quando a entrada de reset R for habilitada, o contador será zerado, isto é, CV=0.

A contagem permanece até a saída CV=PV. Atingindo este valor, o contador permanecerá parado mesmo com pulsos na entrada CU até que seja resetada (R) a saída CV.

Durante a contagem é possível à mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua incrementando a saída CV até CV=PV caso o novo valor de PV seja maior que CV. Sendo o novo valor do preset PV menor que CV, então a contagem é interrompida e a saída Q acionada.



E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	CU	%I, %Q, %M, %MR, %S	Sinal de contagem (pulso).
	R	%I, %Q, %M, %MR, %S	Reset de contagem (CV = 0).
	PV	Constante	Preset de contagem.
SAÍDA	Q	%Q, %M, %MR	Saída do contador.
	CV	%Cx.V	Efetivo de contagem.

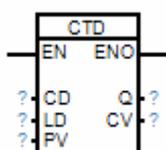
3.3.7 Contador Decrescente (CTD)

A cada pulso em CD a saída CV será decrementada. Inicialmente deve-se ativar LD para carregar o valor a ser decrementado.

Quando o valor de contagem atingir o valor "0" zero a saída Q será ligada, caso contrário permanecerá desligada. A contagem permanece até a saída CV atingir o valor CV=0, o contador permanecerá parado mesmo com pulsos na entrada CD até que seja carregado (LD) um novo valor em CV.

Quando a entrada load (LD) for habilitada, será carregado na saída de contagem (CV) o valor do preset (PV). Tanto PV quanto CV devem ser do mesmo tipo de dado.

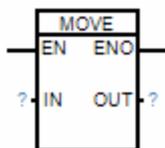
Durante a contagem é possível à mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua decrementando a saída CV até CV=0. Caso a entrada LD seja acionada durante a contagem, será carregado automaticamente o novo valor do preset (PV) para a saída CV e o bloco funcional continua a contagem a partir do novo valor de CV.



E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	CD	%I, %Q, %M, %MR, %S	Sinal de contagem (pulso).
	LD	%I, %Q, %M, %MR, %S	Carrega valor para acumulador.
	PV	Constante	Preset de contagem.
SAÍDA	Q	%Q, %M, %MR	Saída do contador.
	CV	%Cx.V	Efetivo de contagem.

3.3.8 Movimentação de Dados (MOVE)

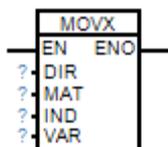
Quando a entrada EN for habilitada, a instrução será executada movendo o conteúdo de IN para OUT.



E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR, %S	Habilita execução da instrução.
	IN	%MB, %MBR, %MW, %MWR, %MD, %MDR, %IW, %QW, %B, %TX.V, %CX.V, Constante	Origem do Dado
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	OUT	%MB, %MBR, %MW, %MWR, %MD, %MDR, %QW	Destino do dado.

3.3.9 Movimentação de Dados com Indexador (MOVX)

Quando a entrada EN for habilitada, caso DIR esteja em 0 o conteúdo da matriz na posição IND será movido para VAR, caso DIR esteja em 1 o conteúdo de VAR será movido para a matriz na posição indicada por IND.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR, %S	Habilita execução da instrução.
	DIR	Constante	0 - Move conteúdo MAT [IND] -> VAR 1 - Move o conteúdo VAR -> MAT [IND]
	MAT*	%MB, %MBR, %MW, %MWR %MD, %MDR %B, Constante	Endereço da Matriz
	IND	%MB, %MBR, %MW, %MWR %MD, %MDR %B, Constante	Índice da Matriz
	VAR	%MB, %MBR, %MW, %MWR %MD, %MDR %B, Constante	Destino ou origem dos dados, depende do operando DIR.
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.

* A variável **MAT** refere-se à primeira posição de memória que será alocada para a matriz, a quantidade de variáveis deve ser reservada de acordo com o tamanho da matriz a ser usada. Exemplo: **MATRIZ1** com 20 posições:

Declaração de **%MB10 = MATRIZ1**, posições reservadas para uso da matriz **%MB10, %MB11, %MB12, até %MB19**.

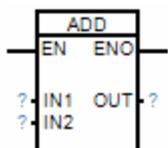
3.3.10 Adição (ADD)

A instrução será executada sempre que a entrada EN estiver habilitada, realizando a expressão:

$$\text{OUT} = \text{IN1} + \text{IN2}$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR, %MW, %MWR, %MD, %MDR, Constante	Origem do Dado 1
	IN2	%MB, %MBR, %MW, %MWR, %MD, %MDR, Constante	Origem do Dado 2
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	OUT	%MB, %MBR, %MW, %MWR, %MD, %MDR	Resultado

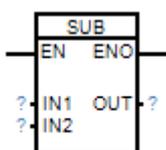
3.3.11 Subtração (SUB)

A instrução será executada sempre que a entrada EN estiver habilitada, realizando a expressão:

$$\text{OUT}=\text{IN1}-\text{IN2}$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR, %MW %MWR, %MD, %MDR, Constante	Origem do Dado 1
	IN2	%MB, %MBR, %MW %MWR, %MD, %MDR, Constante	Origem do Dado 2
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	OUT	%MB, %MBR, %MW %MWR, %MD, %MDR	Resultado

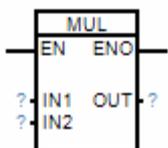
3.3.12 Multiplicação (MUL)

A instrução será executada sempre que a entrada EN estiver habilitada, realizando a expressão:

$$\text{OUT}=\text{IN1}*\text{IN2}$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR, %MW %MWR, %MD, %MDR, Constante	Origem do Dado 1
	IN2	%MB, %MBR, %MW %MWR, %MD, %MDR, Constante	Origem do Dado 2
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	OUT	%MB, %MBR, %MW %MWR, %MD, %MDR	Resultado

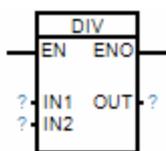
3.3.13 Divisão (DIV)

A instrução será executada sempre que a entrada EN estiver habilitada, realizando a expressão:

$$\text{OUT} = \text{IN1} / \text{IN2}$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

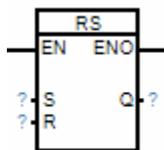


E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR, %MW %MWR, %MD, %MDR, Constante	Origem do Dado 1 (Mesmo tipo de dado)
	IN2	%MB, %MBR, %MW %MWR, %MD, %MDR, Constante	Origem do Dado 2 (Mesmo tipo de dado)
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	OUT	%MB, %MBR, %MW %MWR, %MD, %MDR	Resultado (mesmo tipo de dado de IN1 e IN2)

3.3.14 Rele Biestável (RS)

A instrução será executada sempre que a entrada EN estiver habilitada.

Durante a execução é verificado a alteração do estado de R1 e S sendo que, R1 é dominante sobre S. Quando R1=True a saída Q1 é desacionada independente do estado de S (True ou False). Para acionar a saída Q1: S=True e R1=False

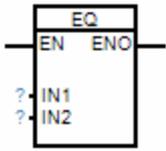


E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	S	%I, %Q, %M, %MR	Entrada Set
	R	%I, %Q, %M, %MR	Entrada Reset
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	Q	%Q, %M, %MR	Saída

3.3.15 Igual (EQ)

Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se o resultado for IGUAL, a saída ENO será ligada, caso contrário será desligada.

$$\text{IN1} = \text{IN2}$$

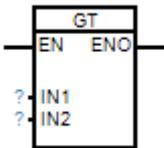


E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR %MW, %MWR, %MD, %MDR, %IW, %B, %TX.V, %CX.V, Constante	Dado 1 (Mesmo tipo de dado)
	IN2	%MB, %MBR %MW, %MWR, %MD, %MDR, %IW, %B, %TX.V, %CX.V, Constante	Dado 2 (Mesmo tipo de dado)
SAÍDA	ENO	%Q, %M, %MR	Saída do comparador

3.3.16 Maior que (GT)

Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se a entrada IN1 for MAIOR QUE IN2, a saída ENO será ligada, caso contrário será desligada.

IN1 > IN2

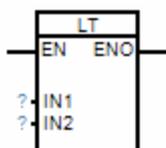


E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR %MW, %MWR, %MD, %MDR, %IW, %B, %TX.V, %CX.V, Constante	Dado 1 (Mesmo tipo de dado)
	IN2	%MB, %MBR %MW, %MWR, %MD, %MDR, %IW, %B, %TX.V, %CX.V, Constante	Dado 2 (Mesmo tipo de dado)
SAÍDA	ENO	%Q, %M, %MR	Saída do comparador

3.3.17 Menor que (LT)

Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se a entrada IN1 for MENOR QUE IN2, a saída ENO será ligada, caso contrário será desligada.

IN1 < IN2

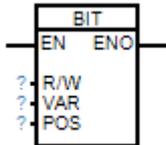


E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR %MW, %MWR, %MD, %MDR, %IW, %B, %TX.V, %CX.V, Constante	Dado 1 (Mesmo tipo de dado)
	IN2	%MB, %MBR %MW, %MWR, %MD, %MDR, %IW, %B, %TX.V, %CX.V, Constante	Dado 2 (Mesmo tipo de dado)
SAÍDA	ENO	%Q, %M, %MR	Saída do comparador

3.3.18 Teste e Set de Bit (BIT)

Quando a entrada EN for habilitada, caso R/W esteja em 1, o bit da posição POS, da variável VAR será testado seu resultado será posto em ENO.

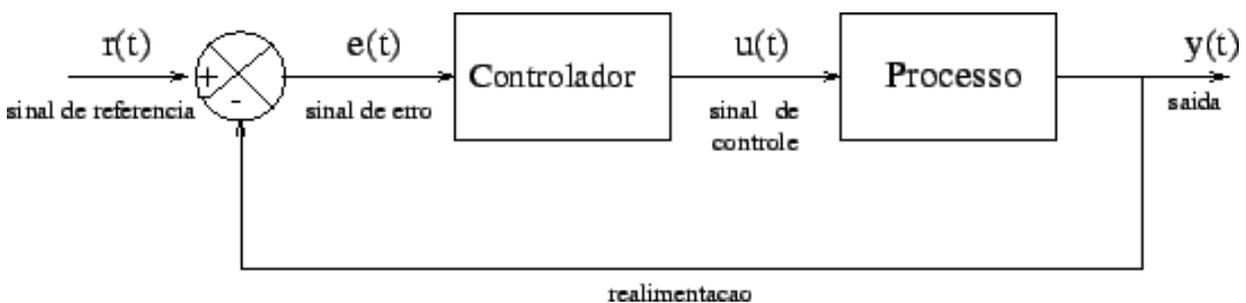
Quando a entrada EN for habilitada, se R/W estiver em 0, será setado o bit, indicado por POS da variável VAR, desabilitando a entrada EN, o bit será resetado, a saída ENO neste caso é uma cópia da entrada.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	R/W	Constante	0 - Seta Bit da variável VAR 1 - Testa Bit da variável VAR
	VAR	%MB, %MBR, %MW, %MWR, %MD, %MDR	Variável a ter seu bit testado ou setado.
	POS	Constante	Posição do bit da variável VAR.
SAÍDA	ENO	%Q, %M, %MR	Saída do comparador

3.3.19 Controle Proporcional, integral, Derivativo (PID)

Sistema de Controle em Malha Fechada no qual a ação de controle depende, de algum modo da saída. Portanto, a saída possui um efeito direto na ação de controle. Neste caso, a saída é sempre medida e comparada com a entrada a fim de reduzir o erro e manter a saída do sistema em um valor desejado.



A realimentação é a característica do sistema de malha fechada que permite a saída ser comparada com a entrada. Geralmente a realimentação é produzida num sistema, quando existe uma seqüência fechada de relações de causa e efeito entre variáveis do sistema.

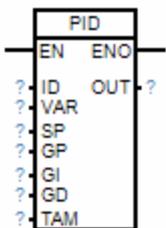
O controlador Proporcional (P), possui uma ação de controle que é proporcional ao erro da saída em relação ao valor desejado.

A lei de controle integral é obtida integrando-se o sinal de erro, ou seja, é equivalente a soma de todos os erros passados.

O controlador derivativo atua somente nas variações do erro, ou seja, prevendo o comportamento do sistema (caráter antecipativo). Por outro lado, o controlador derivativo não pode ser utilizado isoladamente, uma vez que o mesmo não atua quando o erro é constante.

O controlador PID reúne as características de caráter antecipativo e melhoria da estabilidade do controlador PD com as características de erro nulo em regime permanente do controlador PI. A expressão resultante para o controlador PID pode ser dada por:

$$S = K \cdot (e(t) + \sum_0^{\infty} K_i \cdot e(t) \cdot \Delta t + T_d \cdot \Delta e / \Delta t) + \text{BIAS}$$



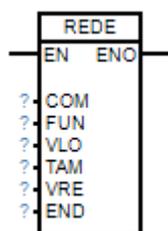
E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	ID	Constante	Identificação do controlador 1 a 8.
	VAR	%MB, %MW, %IW, %QW	Variável do processo
	SP	%MB, %MBR, %MW, %MWR, Constante	Set point do controlador
	GP	%MB, %MBR, %MW, %MWR, Constante	Ganho proporcional
	GI	%MB, %MBR, %MW, %MWR, Constante	Ganho integral
	GD	%MB, %MBR, %MW, %MWR, Constante	Ganho diferencial
	TAM	Constante	Tempo de amostragem
	SAÍDA	ENO	%Q, %M, %MR
OUT		%MB, %MW, %QW	Saída de controle

3.3.20 Comunicação em rede (REDE)

Aviso: Nos sete segundos iniciais após a energização os controladores iniciam com os ajustes padrões nas interfaces de comunicação, com isto é possível bloquear o programa aplicativo a fim de realizar novas configurações. Ajuste padrão:

Protocolo: Modbus escravo, endereço:1, baud: 57.600bps, Interface: RS-232.

O Controlador Programável irá escrever / ler um determinado bloco de variáveis em outro equipamento da rede de comunicação, através do canal serial, ethernet ou via interface *wireless*. O protocolo padrão é o modbus RTU, mestre ou escravo, outros sob consulta.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	COM	Constante	Porta serial do CLP a qual se deseja ler/escrever
	FUN ¹	Constante	Função Modbus
	VLO ²	%M, %MR, %MB, %MBR %MW, %MWR, Constante	Endereço da variável local ou valor a ser lido / escrito
	TAM	Constante	Tamanho do bloco de variáveis a ser lido / escrito
	VRE	Constante	Endereço da variável remota a ser lida / escrita
	END	Constante	Endereço do equipamento remoto.
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.

¹ Funções disponíveis para o protocolo Modbus:

Função	Descrição
1	Leitura de Bit (Read Coil Status)
2	Leitura de Bit (Read Input Status)
3	Leitura de Words (Read Holding Registers)
5	Escrita de Bit (Force Single Coil)
6	Escrita de Word Simples (Preset Single Register)
15	Escrita de múltiplos Bits (Preset Multiples Coils)
16	Escrita de múltiplas Words (Preset Multiples Registers)

² Para o uso de modbus mestre as variáveis disponibilizadas para escrita e leitura em um equipamento modbus escravo são:

- Memórias Bits
- Memórias 8 bits
- Memórias 16 bits

Caso deseje-se escrever outro tipo de variáveis como entradas digitais ou entradas analógicas, as mesmas devem ser movidas para os tipos de variáveis relacionadas acima. Observar o limite e capacidade de memória para cada tipo de CLP.

Endereço Modbus	Endereço CLP	Descrição	Faixa de Valores	Função Modbus
-----------------	--------------	-----------	------------------	---------------

Entradas Digitais				
-------------------	--	--	--	--

0000	%I0	Entrada Digital 1	0 - 1	2
0001	%I2	Entrada Digital 2	0 - 1	2
0002	%I2	Entrada Digital 3	0 - 1	2
...				
0079	%I79	Entrada Digital 80	0 - 1	2

Saídas Digitais				
-----------------	--	--	--	--

0000	%Q0	Saída Digital 1	0 - 1	1,5,15
0001	%Q1	Saída Digital 2	0 - 1	1,5,15
0002	%Q2	Saída Digital 3	0 - 1	1,5,15
...				
0079	%Q79	Saída Digital 80	0 - 1	1,5,15

Memórias Bits				
---------------	--	--	--	--

1000	%M0	Memória bit 1	0 - 1	2,5,15
1001	%M1	Memória bit 2	0 - 1	2,5,15
1002	%M2	Memória bit 3	0 - 1	2,5,15
...				
4999	%M999	Memória bit 4000	0 - 1	2,5,15

Memórias Bits Retentivos				
--------------------------	--	--	--	--

5000	%MR0	Memória bit retentivo 1	0 - 1	2,5,15
5001	%MR1	Memória bit retentivo 2	0 - 1	2,5,15
5002	%MR2	Memória bit retentivo 3	0 - 1	2,5,15
...				
5999	%MR999	Memória bit retentivo 1000	0 - 1	2,5,15

Memórias Bits Sistema				
-----------------------	--	--	--	--

6000	%S0	Memória bit sistema 1	0 - 1	2,5,15
6001	%S1	Memória bit sistema 2	0 - 1	2,5,15
6002	%S2	Memória bit sistema 3	0 - 1	2,5,15
...				
6999	%S100	Memória bit sistema 100	0 - 1	2,5,15

Memórias 8 bits				
-----------------	--	--	--	--

0000	%MB0	Memória byte 1	0 - 255	3,6,16
0001	%MB1	Memória byte 2	0 - 255	3,6,16
0002	%MB2	Memória byte 3	0 - 255	3,6,16
...				
2999	%MB3000	Memória byte 3000	0 - 255	3,6,16

Memórias 8 bits Retentivos				
3000	%MBR0	Memória byte retentivo 1	0 - 255	3,6,16
3001	%MBR1	Memória byte retentivo 2	0 - 255	3,6,16
3002	%MBR2	Memória byte retentivo 3	0 - 255	3,6,16
...				
3999	%MBR1000	Memória byte retentivo 1000	0 - 255	3,6,16

Memórias 16 bits				
4000	%MW0	Memória word 1	0 – 65.535	3,6,16
4001	%MW1	Memória word 2	0 – 65.535	3,6,16
4002	%MW2	Memória word 3	0 – 65.535	3,6,16
...				
6999	%MW3000	Memória word 3000	0 – 65.535	3,6,16

Memórias 16 bits Retentivos				
7000	%MWR0	Memória word retentivo 1	0 – 65.535	3,6,16
7001	%MWR1	Memória word retentivo 2	0 – 65.535	3,6,16
7002	%MWR2	Memória word retentivo 3	0 – 65.535	3,6,16
...				
7999	%MWR1000	Memória word retentivo 1000	0 – 65.535	3,6,16

Entradas Analógicas				
8000	%IW0	Entrada Analógica 1	0 – 4095	3
8001	%IW1	Entrada Analógica 2	0 – 4095	3
8002	%IW2	Entrada Analógica 3	0 – 4095	3
...				
8499	%IW60	Entrada Analógica 500	0 – 4095	3

Saídas Analógicas				
8500	%QW0	Saída Analógica 1	0 – 4095	3,6,16
8501	%QW1	Saída Analógica 2	0 – 4095	3,6,16
8502	%QW2	Saída Analógica 3	0 – 4095	3,6,16
...				
8699	%QW20	Saída Analógica 200	0 – 4095	3,6,16

Bytes do Sistema				
8700	%B0	Bytes Sistema 1	0 – 255	3
8701	%B1	Bytes Sistema 2	0 – 255	3
8702	%B2	Bytes Sistema 3	0 – 255	3
...				
8720	%B20	Bytes Sistema 20	0 – 255	3

Acumuladores Timer's				
9000	%T0	Acumulador Timer 1	0 – 65.535	3
9001	%T1	Acumulador Timer 2	0 – 65.535	3
9502	%T2	Acumulador Timer 3	0 – 65.535	3
...				
9499	%T500	Acumulador Timer 500	0 – 65.535	3

Acumuladores Contadores				
9500	%C0	Acumulador Contador 1	0 – 65.535	3
9501	%C1	Acumulador Contador 2	0 – 65.535	3
9502	%C2	Acumulador Contador 3	0 – 65.535	3
...				
9999	%C500	Acumulador Contador 500	0 – 65.535	3

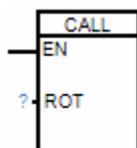
3.3.21 Chama Subrotina (CALL)

Esta instrução realiza o desvio do processamento da rotina corrente para rotina especificada nos seus operandos.

Ao final da execução da rotina chamada, o processamento retorna para a instrução seguinte.

Também é possível chamada uma subrotina dentro de outra subrotina.

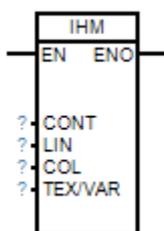
Podem ser usadas até 50 (cinquenta) sub-rotinas em um controlador.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR, %S	Habilita execução da instrução.
	ROT	Nome	Nome da rotina a ser chamada.

3.3.22 IHM (Interface Homem Máquina)

A instrução IHM é utilizada em conjunto com a IHM *Easy View*, toda a programação é realizada de forma integrada ao CLP através do SION tornando a programação e visualização de textos e variáveis do CLP na IHM muito fácil e prática, já que não necessita de um software específico para a mesma.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR, %S	Habilita execução da instrução.
	CONT	Constante	Controle do display da IHM: 0 – mostra variável no formato X 1 – mostra variável no formato X.X 2 – mostra variável no formato X.XX 3 – mostra variável no formato X.XXX 9 – mostra texto sem limpar display 100 a 255 – limpa display e mostra texto (esta faixa de valores é usada para iniciar e identificar diferentes telas)
	LIN	Constante	Indica a posição da linha do cursor do display da IHM faixa de valores 1 a 4
	COL	Constante	Indica a posição da coluna do cursor do display da IHM faixa de valores 1 a 20
	TEX/VAR	%MB, %MBR, %MW, %MWR, %MD, %MDR, %IW, %QW, %B, %TX.V, %CX.V, Constante, %T	Texto ou variável a ser mostrada no display da IHM
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.

Observações quanto à utilização da IHM Easy View:

- Sempre que iniciar uma nova tela usar o controle (CONT) com valores seqüenciais iniciando em 100 (limpa display e mostra texto). Exemplo:

CONT=100 tela 1, limpa display e mostra texto inicial da tela 1

CONT=101 tela 2, limpa display e mostra texto inicial da tela 2

CONT=102 tela 3, limpa display e mostra texto inicial da tela 3

...

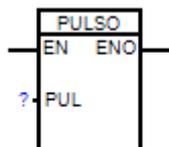
CONT=255 tela 156, limpa display e mostra texto inicial da tela 156.

- O texto a ser mostrado no display dever ser inserido no campo onde se declara a variável do tipo texto (%T) e poderá possuir até 20 caracteres cada texto.

Para mais detalhes consulte o manual específico da IHM Easy View

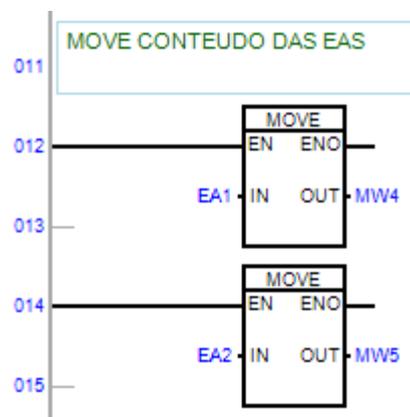
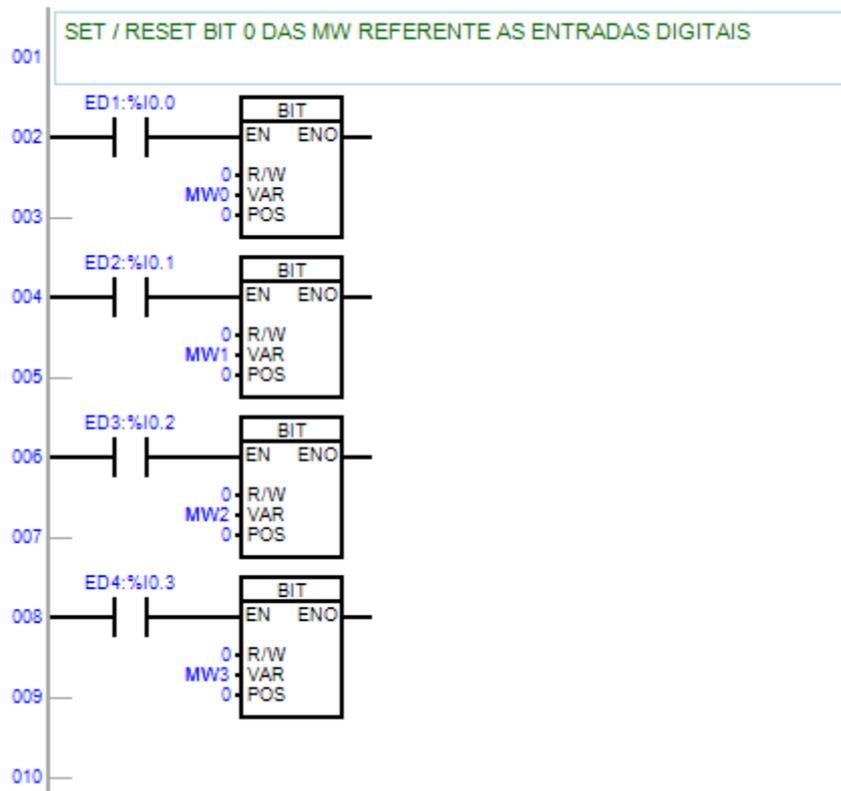
3.3.23 Pulso

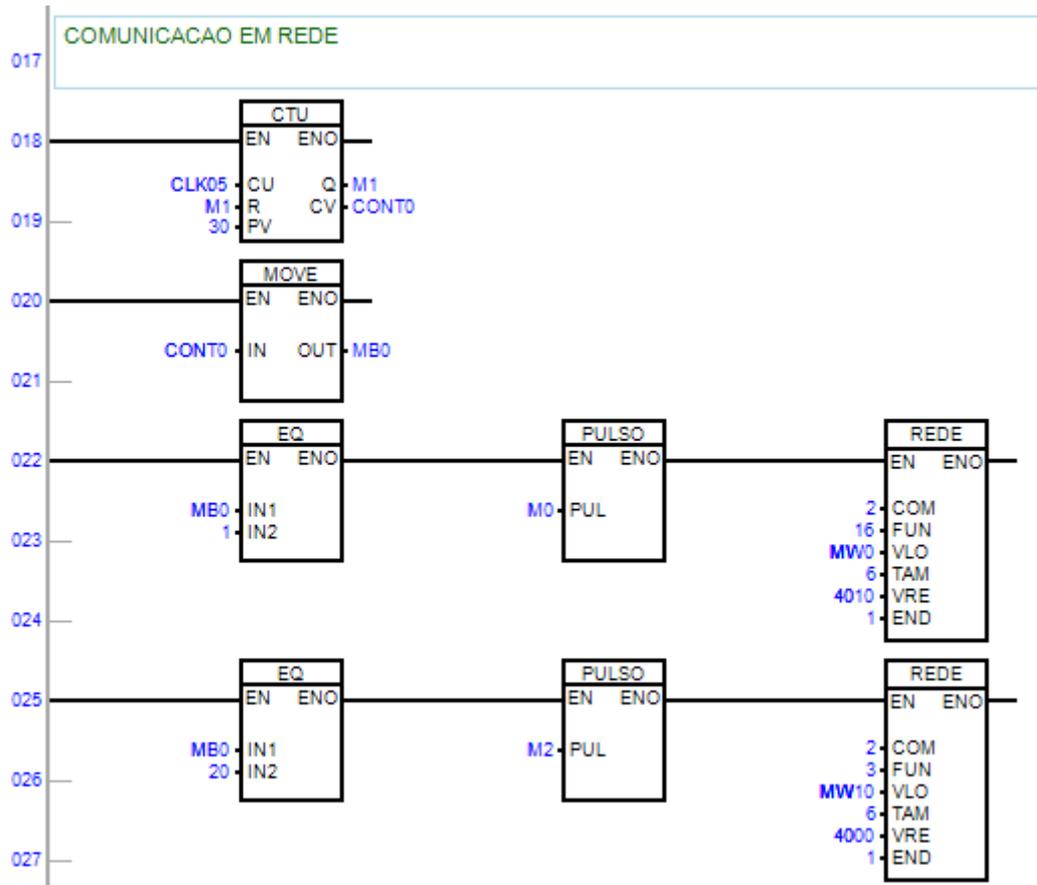
O bloco de função PULSO quando habilitado (EN=1) gera um pulso na saída ENO com o período de um ciclo de varredura. É usado associado a outros blocos funcionais quando se deseja que o bloco ou bobina associado a ele seja executado uma única vez ao acionar sua entrada. Uma aplicação típica para este bloco seria associado ao bloco REDE, aonde deseja-se executar um único evento de comunicação por acionamento do bloco (na transição de desligado para ligado).



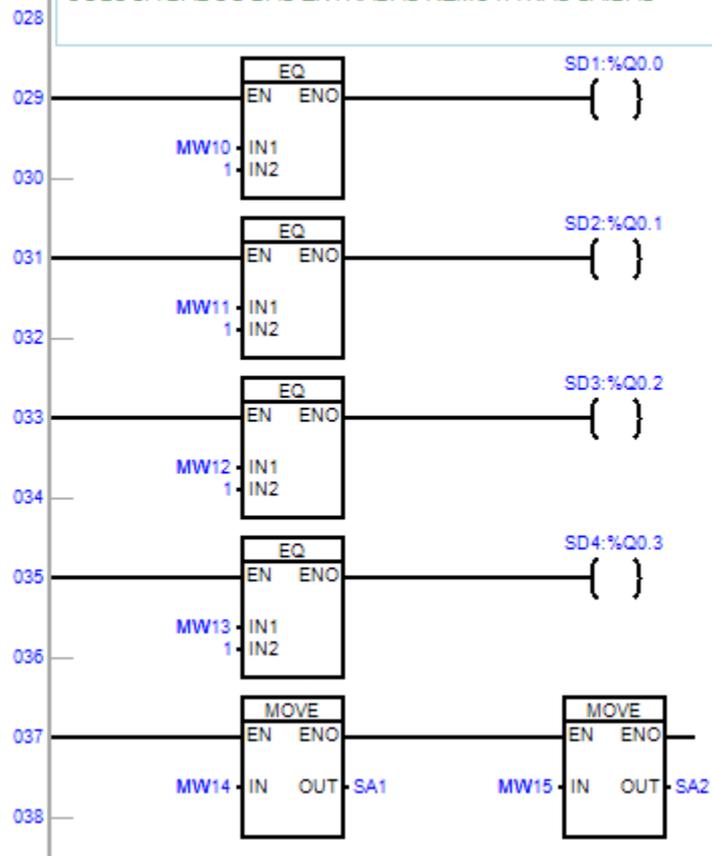
E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR, %S	Habilita execução da instrução.
	PUL	%M, %MR	Memória bit usada para uso interno da instrução, para cada instrução deve ser usado um operando diferente.
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.

EXEMPLOS





COLOCA DADOS DAS ENTRADAS REMOTA NAS SAIDAS



GARANTIA

1 - A INFINIUM garante seus equipamentos contra defeitos de fabricação pelo prazo de 12 (doze) meses contados a partir da data da emissão da nota fiscal.

2 - A garantia compreende o conserto ou substituição, a nosso critério, dos equipamentos desde que efetivamente constatado o defeito.

3 - Para a efetivação da garantia, a INFINIUM deve receber em sua fábrica os equipamentos em questão. Após o conserto os mesmos estarão disponíveis ao cliente na fábrica. Fica por conta do cliente, responsabilidade e despesas de transporte destas mercadorias.

4 - Os equipamentos deverão ser enviados a INFINIUM acompanhados de nota fiscal e um relatório contendo os problemas detectados pelo cliente.

5 - A garantia perde seu efeito quando:

- Os equipamentos forem violados ou sofrerem alterações sem autorização expressa por escrito pela INFINIUM.

- Os equipamentos não forem instalados seguindo rigorosamente as instruções do manual técnico.

- Os equipamentos sofrerem acidentes ou danos provocados por agentes externos.

6 - A garantia não é válida para:

- Defeitos provocados por mau uso ou instalação inadequada dos equipamentos.

- Danos ocasionados por agentes externos tais como inundações, terremotos, tempestades elétricas, problemas de rede elétrica de alimentação, vibrações excessivas, altas temperaturas e quaisquer outros que estejam fora das condições normais de armazenamento, transporte e uso deste equipamento.

- Danos ocasionados a máquinas, processos e pessoais, ocasionados por mau funcionamento destes equipamentos.

7 - A garantia é expressa em termos de performance dos equipamentos de acordo com suas características técnicas expressas claramente no manual. Não compreende, portanto a garantia de performance do sistema onde são empregados os equipamentos INFINIUM, ficando esta ao encargo do engenheiro responsável pelo projeto deste sistema. Por sistema entende-se o conjunto "equipamento eletrônico INFINIUM, sensores e transdutores, acionamentos e mecânica, etc".

8 - A INFINIUM não se responsabiliza por quaisquer outros termos de garantia que não os expressos aqui.