



---

**CONTROLADOR PROGRAMÁVEL *Conn-FLEX***  
**MANUAL DO *HARDWARE* E *SOFTWARE***

---

**Versão 1.17**  
**Julho de 2011**

---

**Infinium Automação Industrial Ltda**  
site: [www.infiniumautomacao.com.br](http://www.infiniumautomacao.com.br)  
email: [contato@infiniumautomacao.com.br](mailto:contato@infiniumautomacao.com.br)

## 1. APRESENTAÇÃO

A linha de Controladores Programáveis *Conn-FLEX* foi projetada para atender os requisitos de automação e controle em máquinas e processos industriais com o grande diferencial de incorporar em um único equipamento, controlador programável em linguagem *ladder* de acordo com a norma IEC-61131-3, interfaces de comunicação serial, transceptor de rádio e interface ethernet, permitindo ampla conectividade para processos distribuídos e dispersos em grandes áreas geográficas ou em parques industriais.

### Características de Destaque

- **Alta conectividade incorpora até quatro interfaces de comunicação diferentes em uma única CPU:**
  - RS-232
  - RS-485
  - Interface Wireless
  - Ethernet
- **Sua estrutura modular aplica-se com excelente performance tanto em sistema de Telecomando e Telemetria como em automação de máquinas e processos distribuídos.**
- **Elevado poder de processamento através de processadores de alta performance.**
- **Permite expansão das E/S incorporadas a CPU.**
- **Programável em linguagem Ladder de acordo com a norma IEC-61131-3.**
- **Protocolos Modbus RTU Mestre / Escravo e Modbus/TCP, selecionáveis via software.**
- **Relógio e Calendário com alta precisão.**

## 2. APLICAÇÕES DOS CONTROLADORES PROGRAMÁVEIS

O Controlador Lógico Programável é um equipamento extremamente versátil, com aplicações em todos os segmentos industriais. Suas características permitem que ele efetue desde simples lógicas até sofisticados controles de processos.

Existe para automatizar processos industriais, sejam de sequenciamento, intertravamento, controle de processos, batelada, etc. Este equipamento tem seu uso tanto na área de automação da manufatura, de processos contínuos, elétrica, predial, dentre outras, como pode ser visto no exemplo que segue:

AUTOMAÇÃO DE MÁQUINAS	CONTROLE DE PROCESSOS
Injetoras de plástico	Saneamento
Extrusoras	Medição e controle de energia
Prensas e retíficas	Químicos
Furadeiras	Siderúrgicos
Robôs e manipuladores	Metalúrgicos
Misturadores	Estufas e secadoras
Máquinas especiais	Supervisão de plantas industriais

### 3. COMUNICAÇÃO

A ampla conectividade do *Conn-FLEX* permite a automação de processos distribuídos de forma simples e amigável, podendo ser executada através de conexões *wireless* e através de interfaces RS-232, RS-485 e ETHERNET.

A utilização do protocolo aberto Modbus mestre ou escravo em todas as interfaces de comunicação facilita a integração com equipamentos de outros fabricantes.

## 4. CARACTERÍSTICAS TÉCNICAS

### 4.1 CPU

O módulo CPU realiza as funções de processamento e controle do sistema. Neste módulo estão localizados os processadores principais, memória *FLASH*, memória RAM, interfaces de comunicação serial, Transceptor de Rádio. Seu núcleo multiprocessado de 16 bits composto por processadores tipo DSP permite a execução de diversas tarefas em paralelo com alta performance e em tempo real.

UNIDADE CENTRAL	
Processador Principal	DSP ( <i>Digital Signal Processor</i> ), 16 bits, 90 MHz
Instruções por segundo	30 Milhões
Barramento de dados	800 Kb/s
Memória Sistema operacional	1.2 Mbits
Memória de Configuração	64 Kbits
Memória de Variáveis	64 Kbits
Memória Programa do Usuário	8 Mbits
Número de módulos de E/S	<i>Consultar</i>
Interface Serial	COM1 Programável RS-232 ou RS-485 COM2 Configurável RS-232, RS-485, <i>Wireless, Ethernet</i>
Interface Ethernet (opcional)	10/100 Base-T (Auto detectável) Protocolo TCP/IP e UDP/IP
Protocolos	Modbus RTU Mestre / Escravo de 1,2 Kbps a 57.600 Kbps
Indicadores	Led ALIM alimentação presente Led TX/RX COM1 Led TX/RX COM2 Led LINK RF Led EXE CPU Executando / Bloqueada
Relógio / Calendário	Erro máximo 15ppm +/- 0,7 segundos / dia Amplitude do calendário: 2000 a 2099 (em anos)
Watchdog	Controlado pelo ciclo de varredura do CP, com período máximo de 1,0 s.
TRANSCÉPTOR DE RÁDIO (INTERFACE <i>WIRELESS</i> , opcional)	
Potencia RF	Até 1.000mW (+30dBm)
Alcance	Até 30Km *
Baud Rate	1200 a 57.600 Kbps
Sensibilidade do receptor	-110 dBm
Faixa de operação	915-928 MHz
Tecnologia	<i>Frequency Hopping Spread Spectrum</i> (Espalhamento Espectral por Saltos de Freqüência)
Topologia de rede	Ponto a ponto e ponto-multiponto
Número de canais	50
Segurança	Identificador ID de um byte. Chave de encriptação de 56 bits.
Impedância da Antena	50 $\Omega$
Conexão Antena	SMA Fêmea

\* com visada em campo aberto, de acordo com projeto teórico de rádio enlace.

**ENTRADAS DIGITAIS (INTEGRADAS A CPU PRINCIPAL)**

Quantidade	8 (oito)
Tipo	P (nível alto ativo com entrada positiva)
Tipo de Entradas	24Vcc Optoisolada
Corrente de Entrada	7 a 10 mA
Isolamento	1500 Vcc
Tensão para nível 0	Abaixo de 5V
Tensão para nível 1	Acima de 10V

**SAIDAS DIGITAIS (INTEGRADAS A CPU PRINCIPAL)**

Quantidade	8 (oito)
Tipo de Saídas	Relê contato NA
Corrente máxima nos contatos	2 A
Tensão máxima aplicável	220Vcc / 250Vca
Numero de operações (mínimo)	2 milhões @1A / 30V

**ENTRADAS ANALÓGICAS (INTEGRADAS A CPU PRINCIPAL)**

Quantidade	6 (seis)
Tipo de Entrada	0-20 mA ou 4-20 mA
Resolução	12 bits (0 a 4095)
Tensão máxima sem dano	30 V (diferencial)
Corrente máxima sem dano	150 mA
Impedância	165Ω (máximo)
Precisão	Melhor que 0,10% do fundo de escala
Proteção	Contra inversão de polaridade
Filtragem	Filtro RC

**SAÍDAS ANALÓGICAS (INTEGRADAS A CPU PRINCIPAL)**

Quantidade	2 (duas)
Tipo	4-20 mA
Impedância máxima	350 Ω (tensão máxima de 7V)
Resolução	12 bits (0 a 4095)
Precisão	1% para o fundo de escala

**GERAIS**

Alimentação	12 a 30 Vcc
Consumo	Máx. 400 mA (em 24 Vcc com transmissão de RF).
Temperatura de Operação	0 a +60 °C
Umidade	10% a 90% (não condensante)
Dimensões	100 X 75 X 105 mm (L X A X P)
Peso	300 g
Material Gabinete	ABS com fixação para trilho DIN

## 5. INSTALAÇÃO

O conjunto de instruções a seguir define os principais pontos que o usuário deve observar na instalação dos Controladores Programáveis *Conn-FLEX*:

- a) Os Controladores devem ser instaladas em uma caixa ou painel que possua vedação completa contra poeira, respingos de água, óleo e produtos corrosivos. Esta caixa ou painel também deve protegê-lo contra choques mecânicos, vibrações mecânicas e altas temperaturas (acima de 60 °C).
- b) Deve-se evitar que os equipamentos sejam montados no mesmo painel de transformadores, contadores, solenóides ou outros componentes eletromecânicos que possam produzir ruídos eletromagnéticos.
- c) Deve ser escolhida uma rede de alimentação isenta de ruído e com o mínimo de flutuação possível para a alimentação dos Controladores.
- d) Evitar a passagem dos cabos de RF, dados e alimentação próximos a cargas indutivas como motores, contadores, solenóides, válvulas.

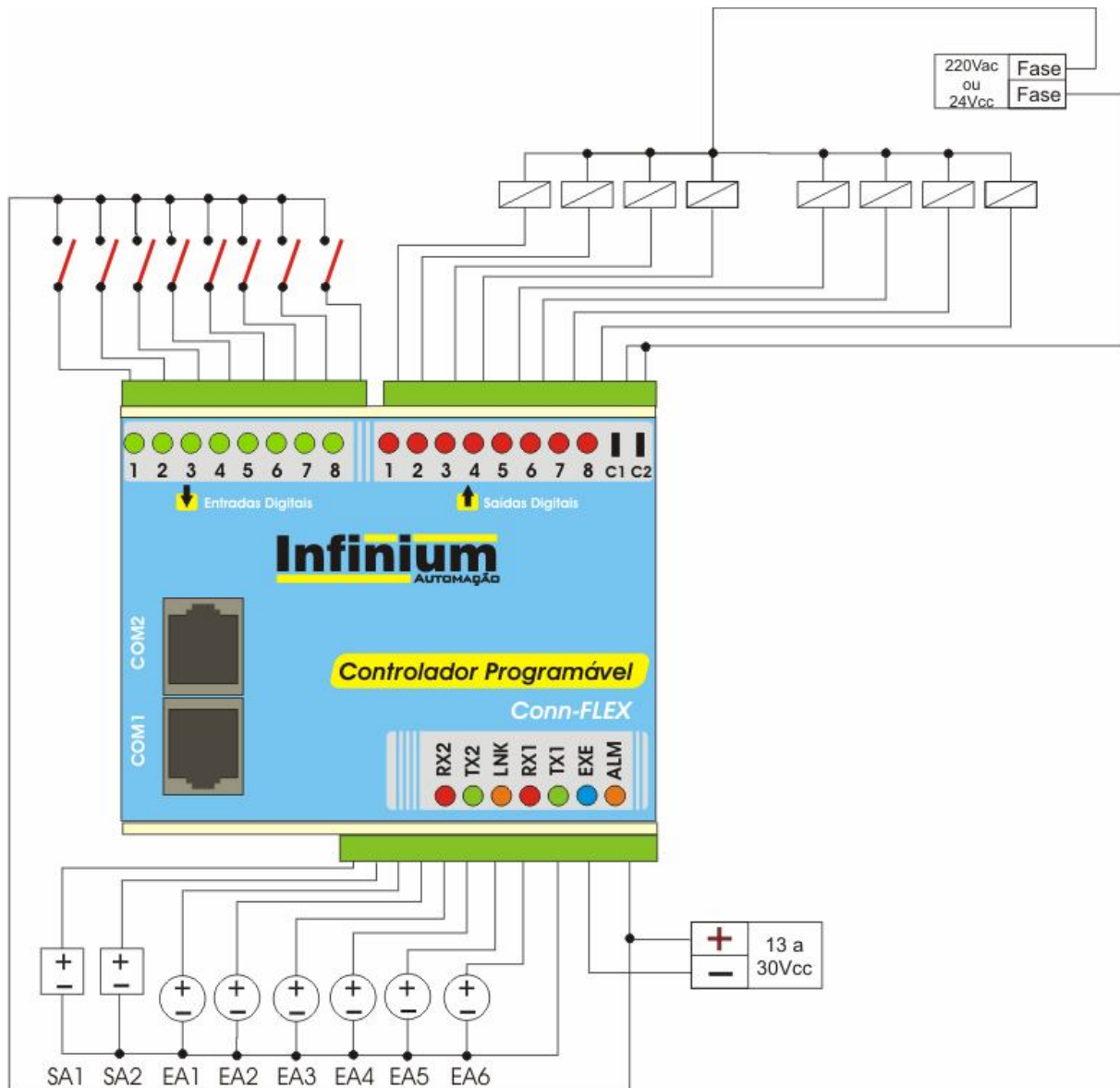
### 5.1 DIAGRAMA DE INTERLIGAÇÃO

Os Controladores *Conn-FLEX* aceitam alimentação na faixa de 13 a 30 Vcc, a mesma deverá garantir boa filtragem e alta imunidade a ruídos, a ligação é feita de acordo com a ilustração que segue.

DESCRIÇÃO DOS BORNES	
<b>ED1 a ED8</b>	Entradas Digitais 1 a 8
<b>SD1 a SD8</b>	Saídas Digitais 1 a 8 (Contato NA)
<b>C1</b>	Comum 1 (Comum das saídas 5 a 8)
<b>C2</b>	Comum 2 (Comum das saídas 1 a 4)
<b>EA1 a EA6</b>	Entradas Analógicas 1 e 6
<b>SA1 a SA2</b>	Saídas Analógicas 1 e 2
<b>+V<sup>1</sup></b>	Alimentação - Positivo (+)
<b>0V<sup>2</sup></b>	Alimentação - Negativo (-)

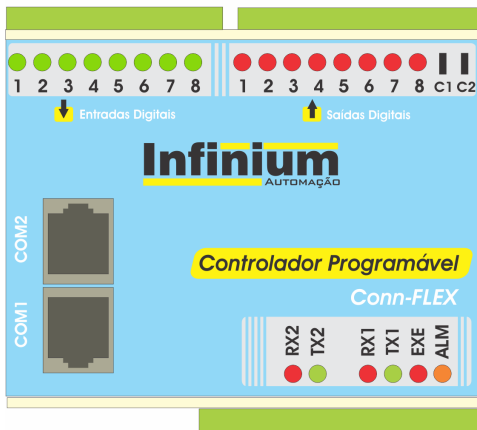
<sup>1</sup> *Existem dois bornes +V, estes são interligados internamente podendo ser usado como comum das Entradas Digitais.*

<sup>2</sup> *Existem dois bornes 0V, estes são interligados internamente podendo ser usados como comum das entradas e saídas analógicas.*

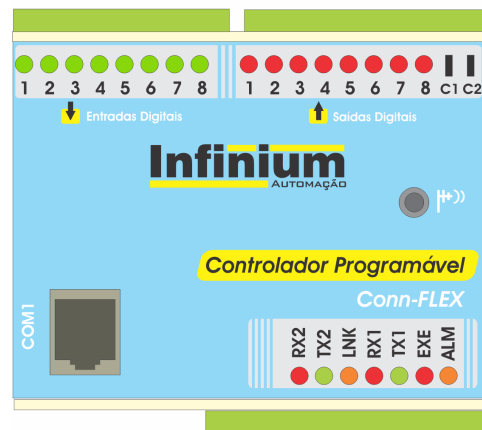


## 5.2 INTERFACES DE COMUNICAÇÃO

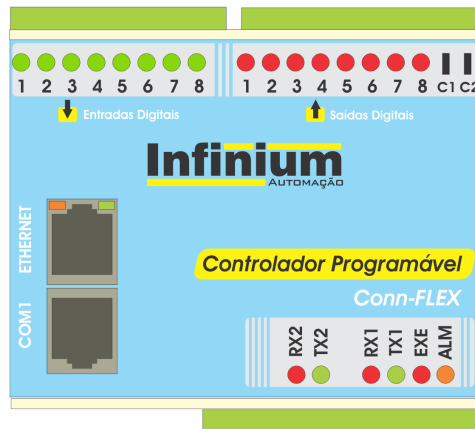
O Controlador Programável *Conn-FLEX* possui duas interfaces de comunicação serial, podendo ser configuradas como RS-232 ou RS-485, via software. Opcionalmente no lugar da COM2 poderá ser fornecido com interface *Wireless* e ainda Ethernet. A interface serial segue a norma EIA232 e EIA485, com relação aos níveis de tensão sua conexão é feita através de um conector RJ-45 fêmea de acordo com a pinagem que segue. A programação pode ser executada através das portas COM1 ou COM2, ou ainda via Ethernet quando equipado com este opcional.



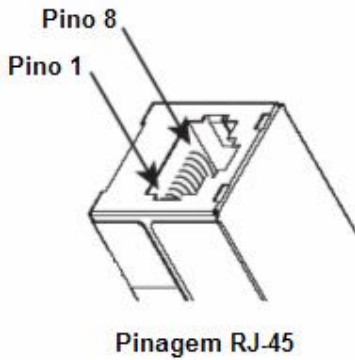
**Interface RS232/RS485**



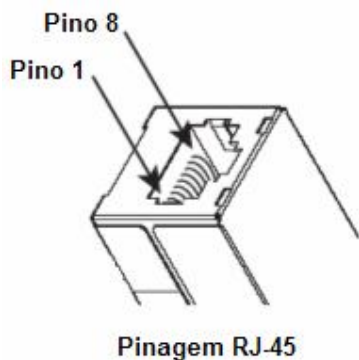
**Interface via rádio**



**Interface ETHERNET**



PINAGEM CONECTOR SERIAL RJ-45		
PINO	DESCRIÇÃO	SENTIDO
1	D- RS485	-
2	RXD – Recepção de dados RS232	Entrada
3	TXD – Transmissão de Dados RS232	Saída
4	D+ RS485	-
5	GND	-
6	Não Conectado	-
7	RTS - Request to send RS232	Saída
8	CTS - Clear to send RS232	Entrada



PINAGEM CONECTOR ETHERNET RJ-45		
PINO	DESCRIÇÃO	SENTIDO
1	TX Diferencial +	Saída
2	TX Diferencial -	Saída
3	RX Diferencial +	-
4	Não Conectado	Entrada
5	Não Conectado	-
6	RX Diferencial -	Entrada
7	Não Conectado	-
8	Não Conectado	-

**Aviso:** Nos sete segundos iniciais após a energização o controlador inicia 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.

## 6. INDICADORES

O painel frontal possui Led's indicadores que tem as seguintes funções:

LED		FUNÇÃO
CPU	Alim	Alimentação Presente, Controlador Ligado
	EXE	- <b>Aceso:</b> indica que o programa do usuário encontra-se bloqueado (STOP). - <b>Aceso apagando rapidamente:</b> Inicialização, neste momento o Controlador encontra-se com os ajustes padrões. - <b>Piscando:</b> programa do usuário sendo executado (RUN). - <b>Piscando intermitentemente e rapidamente:</b> memória FLASH vazia ou em falha.
RF <sup>1</sup>	LNK	Quando configurado como Servidor sempre acesso, quando configurado como Cliente só acende quando conectado a um Servidor.
COM1	TX	Indica transmissão de dados
	RX	Indica recepção de dados
COM2	TX	Indica transmissão de dados
	RX	Indica recepção de dados
ETHERNET <sup>1</sup>	10	Conexão ethernet 10Mbps
	100	Conexão ethernet 100Mbps
	COL	Indica Colisão de dados na rede
	LNK	Link Ethernet
ENTRADAS	ED1 a ED8	Sinalização de entradas digitais ligadas
SAIDAS	SD1 a SD8	Sinalização de saídas digitais ligadas

<sup>1</sup> Quando equipado com este opcional.

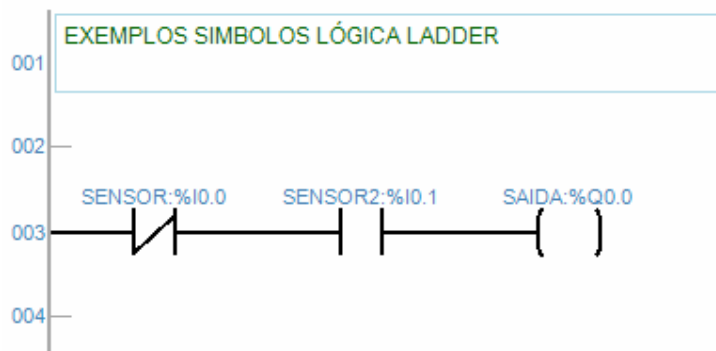
## 7. PROGRAMAÇÃO

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.

### 7.1 OPERANDOS E ENDEREÇOS

#### 7.1.1 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.

**%I0.1 - ponto 1 do octeto de entrada 0**

### **7.1.2 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.

***%Q1.2 - ponto 2 do octeto de saída 1***

### **7.1.3 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)***

Faixa de valores %M0 a %M1599 – Total de 1600 memórias.

### **7.1.4 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)***

Faixa de valores %MB0 a %MB799 – Total de 800 memórias.

### **7.1.5 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)***

Faixa de valores %MW0 a %MW399 – Total de 400 memórias.

### 7.1.6 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)***

Faixa de valores %MD0 a %MD099 – Total de 100 memórias.

### 7.1.7 Operandos Timers

#### **%Tx.V – Acumuladores Timer**

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

#### ***%TM2.V – valor do acumulador do timer 2***

Faixa de valores %T0.V a %T119.V – Total de 120 Timer's

### 7.1.8 Operandos Contadores

#### **%Cx.V – Acumuladores Timer**

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

#### ***%CM10.V – valor do acumulador do contador 10***

Faixa de valores %C0.V a %C119.V – Total de 120 Timer's

### 7.1.9 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.

A quantidade de entradas depende do número de módulos instalados.

### **7.1.10 Operandos %QW – Saídas analógicas (Word de saída)**

Os operandos %QW são usados 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”. A quantidade de saídas depende do numero de módulos instalados.

### **7.1.11 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***

***%S8 - Tecla < - > da IHM***

***%S9 - Tecla < ENTER > da IHM***

***%S10 - Tecla < SHIFT > da IHM***

***%S11- Link de RF (0: sem link, 1:link ok, se servidor sempre 1)***

### **7.1.12 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 (caso instalado)***

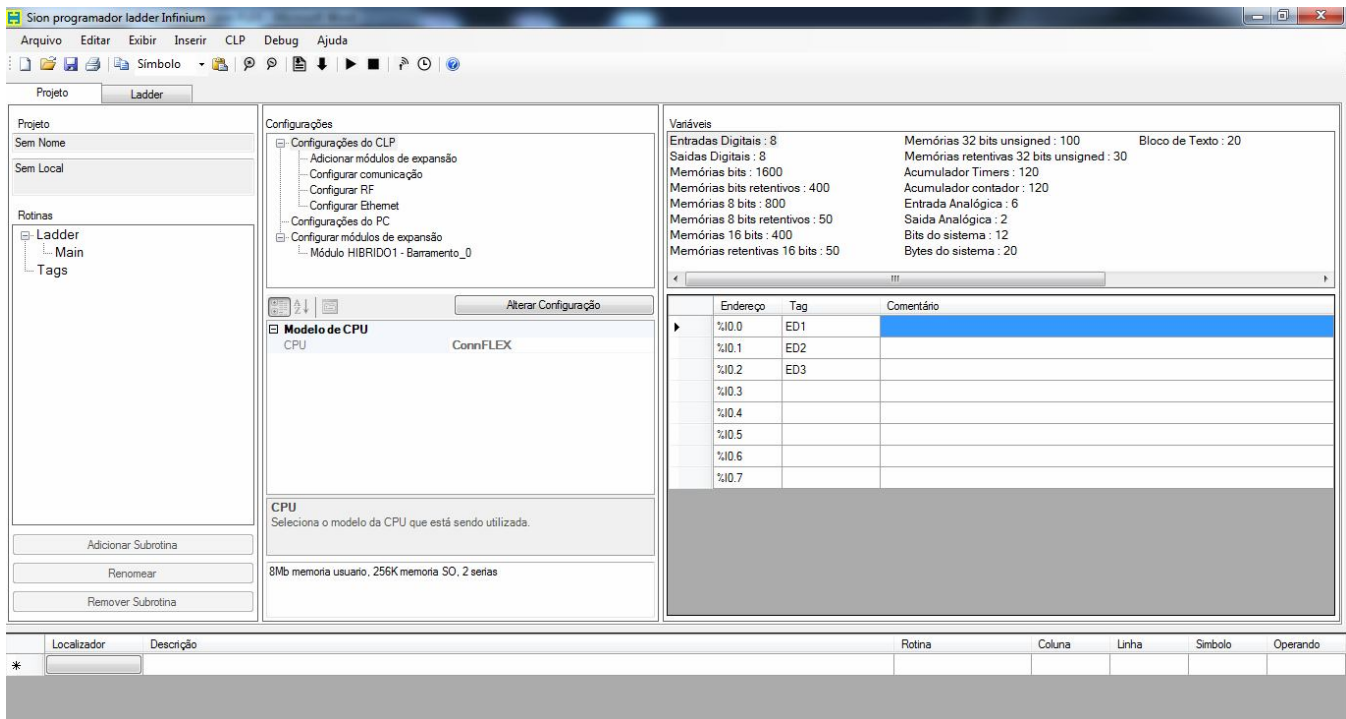
***%B11 - Status de Comunicação ETH (caso instalado)***

### 7.1.13 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.

## 7.2 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 2.0 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.

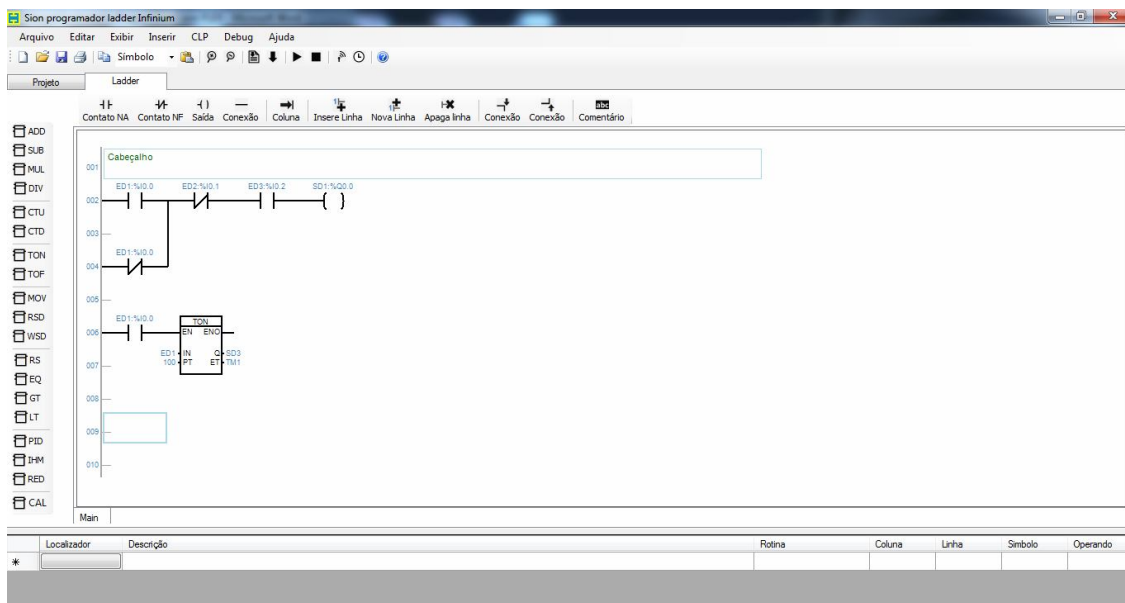
**VARIÁVES:** Nesta área são listadas os operandos / variáveis do CLP, nela pode-se associar nomes aos operandos para facilitar a programação e identificação no programa ladder.

### 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 o manual do CLP.

## 7.3 LADDER

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



### 7.3.1 Contato Aberto

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



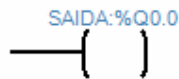
### 7.3.2 Contato Fechado

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



### 7.3.3 Bobinas

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



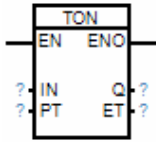
### 7.3.4 Temporizador na Energização (TON)

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

### 7.3.5 Temporizador na Deserregização (TOF)

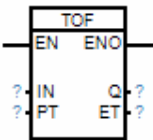
Com as entradas EN e IN habilitadas, na transição de TRUE para FALSE da entrada IN, inicia-se a contagem de tempo na saída ET (Base de tempo décimos de segundo) até o valor de preset (PT). Durante essa contagem, a saída Q permanece ligada.

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

O RESET de temporização acontece quando a entrada IN for ligada, 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 desligada.



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	Desabilitado no fim da temporização
	ET	%Tx.V	Efetivo do temporizador

### 7.3.6 Contador Crescente (CTU)

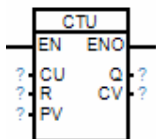
A cada pulso em CU, a saída CV será incrementada.

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	DESCRICAÇÃO
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.

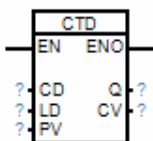
### 7.3.7 Contador Decrescente (CTD)

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.

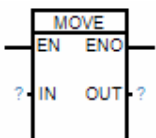
Durante a contagem é possível à mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua decrementando a saída CV até CV=0. Caso a entrada LD seja acionada durante a contagem, será carregado automaticamente o novo valor do preset (PV) para a saída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



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
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.

### 7.3.8 Movimentação de Dados (MOVE)

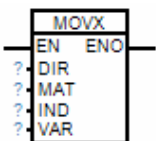
Quando a entrada EN for habilitada, a instrução será executada movendo o conteúdo de IN para OUT.



E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
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	Destino do dado.

### 7.3.9 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.**

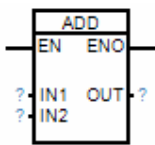
### 7.3.10 Adição (ADD)

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	DESCRICAO
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

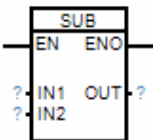
### 7.3.11 Subtração (SUB)

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	DESCRICAO
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

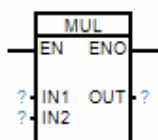
### 7.3.12 Multiplicação (MUL)

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	DESCRICAO
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

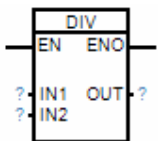
### 7.3.13 Divisão (DIV)

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.

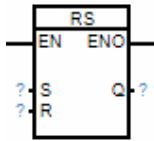


E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR, %MW %MWR, %MD, %MDR, Constante	Origem do Dado 1 (Mesmo tipo de dado)
	IN2	%MB, %MBR, %MW %MWR, %MD, %MDR, Constante	Origem do Dado 2 (Mesmo tipo de dado)
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano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

### 7.3.14 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

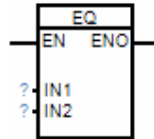


E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	S	%I, %Q, %M, %MR	Entrada Set
	R	%I, %Q, %M, %MR	Entrada Reset
SAÍDA	ENO	%Q, %M, %MR	Cópia do valor booleano de EN.
	Q	%Q, %M, %MR	Saída

### 7.3.15 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.

$$IN1 = IN2$$

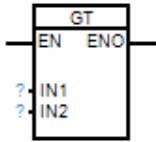


E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR %MW, %MWR, %MD, %MDR, %B, %TX.V, %CX.V, Constante	Dado 1 (Mesmo tipo de dado)
	IN2	%MB, %MBR %MW, %MWR, %MD, %MDR, %B, %TX.V, %CX.V, Constante	Dado 2 (Mesmo tipo de dado)
SAÍDA	ENO	%Q, %M, %MR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

### 7.3.16 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.

$$IN1 > IN2$$

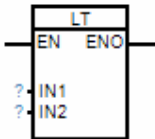


E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR, %MW, %MWR, %MD, %MDR, %B, %TX.V, %CX.V, Constante	Dado 1 (Mesmo tipo de dado)
	IN2	%MB, %MBR, %MW, %MWR, %MD, %MDR, %B, %TX.V, %CX.V, Constante	Dado 2 (Mesmo tipo de dado)
SAÍDA	ENO	%Q, %M, %MR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

### 7.3.17 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.

$$IN1 < IN2$$

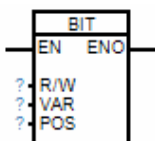


E/S	NOME	TIPO DE DADO	DESCRICAÇÃO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	IN1	%MB, %MBR, %MW, %MWR, %MD, %MDR, %B, %TX.V, %CX.V, Constante	Dado 1 (Mesmo tipo de dado)
	IN2	%MB, %MBR, %MW, %MWR, %MD, %MDR, %B, %TX.V, %CX.V, Constante	Dado 2 (Mesmo tipo de dado)
SAÍDA	ENO	%Q, %M, %MR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

### 7.3.18 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.

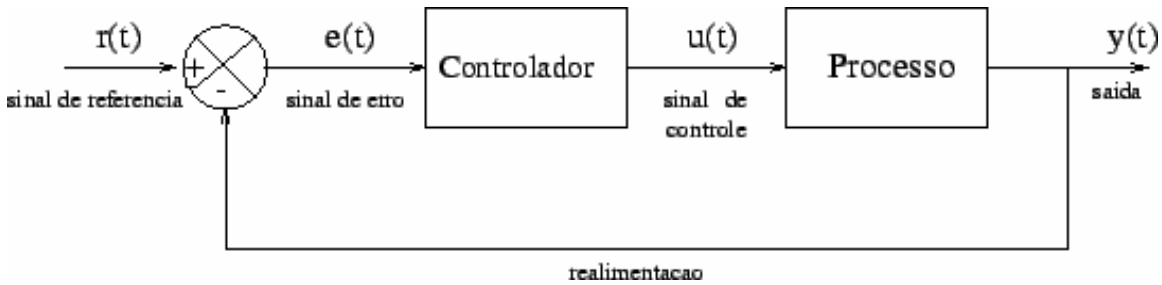
Quando a entrada EN for habilitada, se R/W estiver em 0, será setado o bit, indicado por POS da variável VAR, desabilitando a entrada EN, o bit será resetado, a saída ENO neste caso é uma cópia da entrada.



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

### 7.3.19 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.

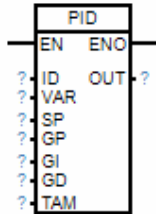
O controlador Proporcional (P), possui uma ação de controle que é proporcional ao erro da saída em relação ao valor desejado.

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:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{d}{dt} e(t) + P \text{ ou então } u(t) = K_p \left( e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + T_d \frac{d}{dt} e(t) \right) + P$$



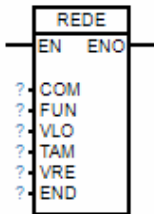
E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	ID	Constante	Identificação do controlador 1 a 8.
	VAR	%MB, %MW, %IW, %QW	Variável do processo
	SP	%MB, %MBR, %MW, %MWR, Constante	Set point do controlador
	GP	%MB, %MBR, %MW, %MWR, Constante	Ganho proporcional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

### 7.3.20 Comunicação em rede (REDE)

**Aviso:** Nos sete segundos iniciais após a energização o controlador inicia 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 ou via interface wireless. O protocolo padrão é o modbus RTU, mestre ou escravo, outros sob consulta.



E/S	NOME	TIPO DE DADO	DESCRICAO
ENTRADA	EN	%I, %Q, %M, %MR	Habilita execução da instrução.
	COM	Constante	Porta serial do CLP a qual se deseja ler/escrever
	FUN <sup>1</sup>	Constante	Função Modbus
	VLO <sup>2</sup>	%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.

<sup>1</sup> 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)

<sup>2</sup> 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	%I0.0	Entrada Digital 1	0 - 1	2
0001	%I0.1	Entrada Digital 2	0 - 1	2
0002	%I0.2	Entrada Digital 3	0 - 1	2
...				
0079	%I7.7	Entrada Digital 80	0 - 1	2

Saídas Digitais				
0000	%Q0.0	Saida Digital 1	0 - 1	1,5,15
0001	%Q0.1	Saida Digital 2	0 - 1	1,5,15
0002	%Q0.2	Saida Digital 3	0 - 1	1,5,15
...				
0079	%Q7.7	Saida 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

<b>Memórias Bits Retentivos</b>				
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

<b>Memórias Bits Sistema</b>				
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

<b>Memórias 8 bits</b>				
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

<b>Memórias 8 bits Retentivos</b>				
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

<b>Memórias 16 bits</b>				
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

<b>Memórias 16 bits Retentivos</b>				
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

<b>Entradas Analógicas</b>				
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

<b>Saídas Analógicas</b>				
8500	%QW0	Saida Analógica 1	0 – 4095	3,6,16
8501	%QW1	Saida Analógica 2	0 – 4095	3,6,16
8502	%QW2	Saida Analógica 3	0 – 4095	3,6,16
...				
8699	%QW20	Saida Analógica 200	0 – 4095	3,6,16

<b>Bytes do Sistema</b>				
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

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

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

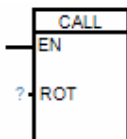
### 7.3.21 Chama Subrotina (CALL)

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.

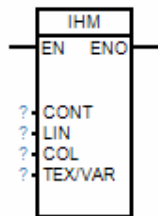
Podem ser usadas até 50 (cinquenta) sub-rotinas em um controlador Conn-FLEX.



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.

### 7.3.22 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

...

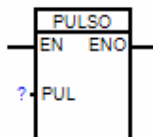
CONT=255 tela 156, limpa display e mostra texto inicial da tela 156.

- 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**

#### 7.3.23 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.

## 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.

## 8. DIAGNÓSTICO E SOLUÇÃO DE PROBLEMAS

### 8.1 MANUTENÇÃO REGULAR E PREVENTIVA

O controlador programável não exige um procedimento obrigatório de manutenção regular, porém, a checagem periódica de alguns itens do sistema contribui para garantir a alta confiabilidade do sistema a longo prazo:

b) Limpeza - Verificar se existe contaminação do controlador programável por poeira, líquidos ou outros produtos. Pode ser necessária uma revisão da vedação da caixa ou painel. Se houver aberturas para ventilação com filtros, estes também devem ser verificados.

c) Temperatura - Verificar se a temperatura ao redor do controlador programável está dentro dos limites adequados. Mesmo a partir de uma instalação original bem feita, as condições podem mudar pela colocação de outros equipamentos nas imediações.

d) Vibração - A instalação do controlador programável em ambientes com vibração mecânica pode provocar problema na fixação dos produtos, conectores, fiação, etc. Estes aspectos devem ser verificados.

e) Ruído - Mesmo que a instalação original seja adequada do ponto de vista da imunidade ao ruído é comum sua modificação, ampliação ou instalação de novos equipamentos no mesmo ambiente. Verificar se os cuidados exigidos para uma boa instalação continuam a serem adotados.

### 8.2 FALHAS

Os controladores programáveis oferecem alguns recursos para diagnósticos de problemas. Estes recursos se baseiam na sinalização através de led's e operandos do sistema.

a) Falha de alimentação – Verificar através do led *Alim*. Verificar a ligação correta da alimentação, o tipo de fonte usado e os níveis de tensão aceitáveis para este tipo de fonte.

b) Falha na comunicação – Pode ser diagnosticada através dos Led's TX, RX, LNK.

c) Falha na execução do programa aplicativo – Pode ser diagnosticada através dos Led's EXE.

### 8.3 REMESSA PARA MANUTENÇÃO

Antes de enviar os equipamentos para manutenção, verificar o sistema em relação aos itens apresentados ao longo deste capítulo. Reunir todas as informações e entrar em contato com o departamento de Assistência Técnica da INFINIUM.

Se houver outro controlador programável da mesma configuração disponível, uma boa alternativa é trocar por outro, a fim de confirmar se é mesmo o controlador programável que está com problema. Neste caso, é sempre importante certificar-se que o controlador programável original não foi danificado por uma tensão incorreta de entrada ou de alimentação.

Confirmada a necessidade de envio do controlador programável para conserto na INFINIUM, enviá-lo como “Remessa para Conserto”. Anexar sempre um relatório onde conste o defeito observado e outras informações julgadas relevantes para facilitar o trabalho e evitar a repetição do problema.