

COTIP
Colégio Técnico Industrial de Piracicaba

APOSTILA 1 C.L.P. – LOGO!

Prof. Anderson Rodrigo Rossi

OBS.: Este curso será baseado em CLP – LOGO!-SIEMENS.

Piracicaba, 26 de janeiro de 2009.

1 – Introdução e História do P.L.C.

C.L.P. - Controlador Lógico Programável definido primeiramente nos E.U.A. como P.L.C. – *Programmable Logic Controller*, são equipamentos responsáveis por aplicações comerciais e industriais.

O primeiro CLP foi desenvolvido no final de 1960. Foi desenvolvido para atender a flexibilidade das indústrias, no caso a automobilística onde a linha de montagem é dinâmica em relação ao modelo do carro a ser produzido. Foi então que a Bedford Associates ofereceu a General Motors uma solução. Tratava-se de um dispositivo que poderia funcionar em várias operações distintas e facilmente programáveis. Esse equipamento era o Modular Digital Controller (MODICON), sendo o MODICON 084 o primeiro modelo comercial, apresentado na Figura 1.1.



Figura 1.1 – MODICON 084 – primeiro modelo comercial de CLP.

Já em 1970, o CLP era equipado com uma CPU, com processador AMD 2901.

Em 1973 surgiu a primeira comunicação entre CLPs – Mod bus.

Em 1980, surge a primeira comunicação Standard – MAP (Manufacturing Automation Protocol).

Em 1990 chega a norma IEC 1131-3 que leva todas as linguagens a um padrão internacional.

Hoje, dentro da nova IEC 61131-3, podemos programar o CLP de quatro modos: diagrama de blocos, lista de instruções, *ladder* e texto estruturado.

Um CLP monitora entradas, toma decisões baseado em uma programação, e controla saídas para automatizar um processo ou máquina. A Figura 1.2 apresenta a integração do equipamento com as entradas e saídas.

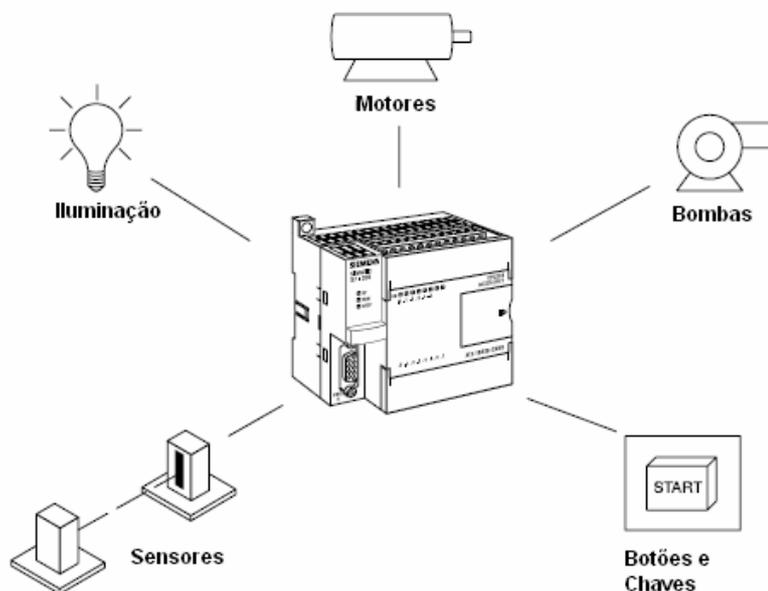


Figura 1.2 – Integração de PLC com entradas e saídas.

O que são entradas? São dispositivos que introduzem informações ao CLP, tais dispositivos são como chaves, botões, sensores, encoders, termopares, PT100, etc.

O que são saídas? São dispositivos que recebem uma informação do CLP para executar uma determinada ação, tais dispositivos são como motores, bombas, cilindros, resistências, etc.

Vantagens do CLP:

- Economia na mudança de função (facilidade de programação);
- Alta confiabilidade;
- Aumento da vida útil do controlador;
- Menor manutenção preventiva e corretiva;
- Envio de dados para processamentos centralizados;
- Expansão em módulos;
- Redução de dimensão em relação a painéis de Relês, para redução de custos.

O funcionamento de um CLP corresponde a três etapas distintas, as quais são: entradas, processamento e saídas. Essas etapas são ilustradas na Figura 1.3.



Figura 1.3 - Estrutura básica de funcionamento de um CLP.

O hardware de um CLP é formado por 3 unidades distintas, as quais são: fonte de alimentação, CPU (Unidade Central de Processamento) e interfaces de entrada e saídas ou I/O, a Figura 1.4 apresenta as unidades em um modelo de micro CLP.

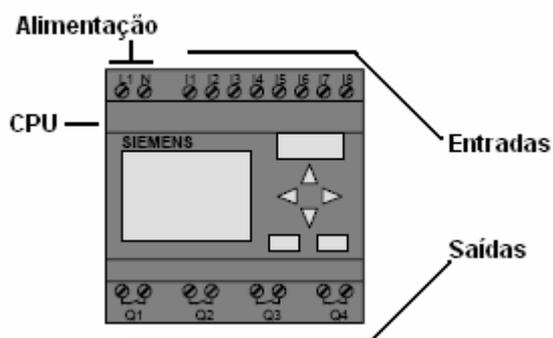


Figura 1.4 – Unidades de hardware em um micro PLC (LOGO! – Siemens).

Tipos de Programação

O que é linguagem de programação?

Uma linguagem de programação é um meio de indicar a um sistema de execução de tarefas uma série de operações a serem executadas. Uma linguagem de programação é, sobretudo, um meio de exprimirmos idéias acerca de metodologias.

Até início de 1990 não existiam técnicas de programação padrão para CLP. Os sistemas utilizados eram baseados em textos estruturados em linguagens como Basic, Fortran, C e várias outras linguagens. A não padronização de linguagem tinha desvantagens como desperdício de tempo, alto custo em treinamento e falta de integração de sistemas.

O International Electro-technical Commission (IEC) iniciou trabalhos para padronizar a programação de PLC e atualmente lança mão da IEC 61131-3.

Suas vantagens são:

- Flexibilidade de programação, através de 3 modos gráficos e dois textos estruturados. O programador pode escolher qual é a melhor linguagem segundo o processo, e até utilizar várias linguagens em um mesmo processo mantendo a integração;

- Permite que o programa seja “quebrado” em elementos funcionais;
- Reduz erros, principalmente aos relacionados com a digitação;
- Sistemas seqüenciais podem ser programados passo a passo através de “sequential function chart” (SFC) – Grafecet.

Os tipos de linguagens de programação para CLPs são:

- Texto Estruturado - É uma linguagem de alto nível, com uma sintaxe que lembra o Pascal, e é indicada para uma ampla faixa de aplicações. Figura 1.5.

Q3 <= (I1 OR I2) AND I3

Figura 1.5 – Exemplo de programação de CLP com texto estruturado.

- Ladder - É uma linguagem baseada na lógica de contatos, como se fosse um comando elétrico na horizontal. Figura 1.6.

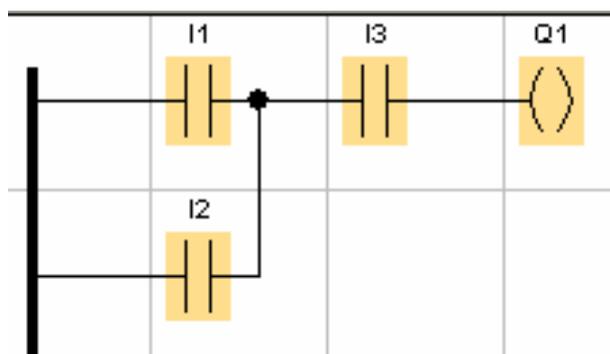


Figura 1.6 – Exemplo de programação de CLP com ladder.

- Lista de Instruções - É uma linguagem de baixo nível, define-se textualmente os componentes e as instruções que o CLP deverá realizar passo a passo. Figura 1.7.

<i>Instruction number</i>	<i>Operation</i>	<i>Address</i>	<i>Comment</i>
00	: LD	I 1	(*Load the value of I1*)
01	: OR	I 2	(*OR with I2*)
02	: AND	I 3	(*AND with I3*)
03	: ST	Q 1	(*Store result to Q1*)

Figura 1.7 – Exemplo de programação de CLP com lista de instruções.

- Diagrama de Blocos - É semelhante ao circuito digital. Temos portas lógicas e blocos montados em caixas selecionáveis. Figura 1.8.

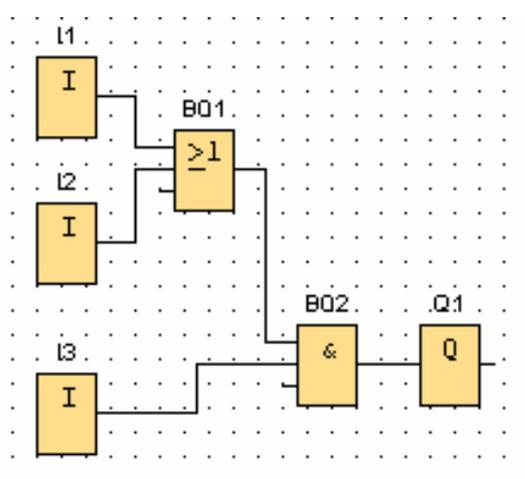


Figura 1.8 – Exemplo de programação de CLP com diagrama de blocos.

- SFC (Sequential Function Chart, conhecido como Grafecet) - É uma linguagem indicada para processos seqüenciais e pode misturar lista de instruções, diagramas ladder e blocos de funções. Figura 1.9.

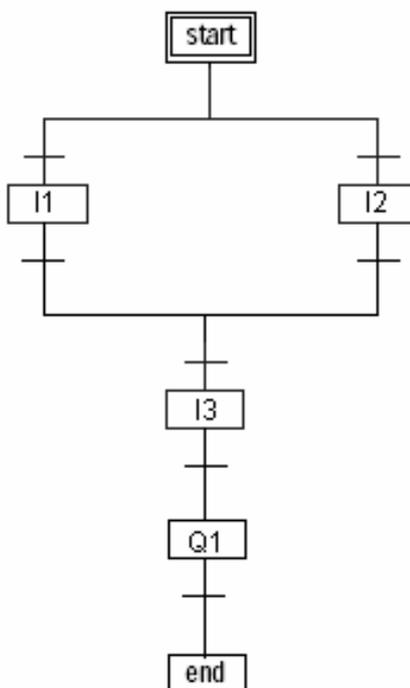


Figura 1.9 – Exemplo de programação de CLP com SFC (Grafcet).

Com relação ao endereçamento (baseado em LOGO! – SIEMENS).

Entradas

Entrada normal	<u>I1, I2, I3, I4, ..., I12.</u>
Entrada Analógica	<u>AI1, AI2.</u>
Entrada ASi (Conexão com Bus ASi)	<u>Ia1, Ia2, Ia3, Ia4.</u>

Saídas

Saída normal	<u>Q1, Q2, Q3, Q4, ..., Q8.</u>
Saída ASi (Conexão com Bus ASi)	<u>Qa1, Qa2, Qa3, Qa4.</u>

Marcador

M1, M2, M3, M4, ..., M8

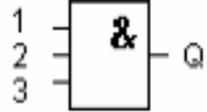
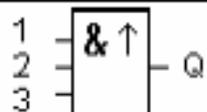
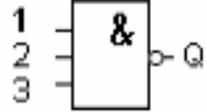
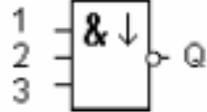
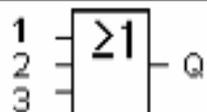
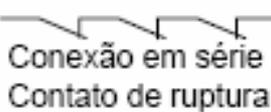
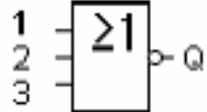
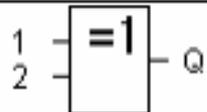
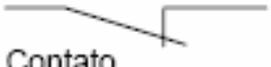
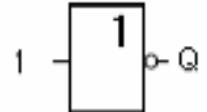
Níveis Fixos

Níveis Fixos

Alto (hi)	<u>hi</u>
Baixo (lo)	<u>lo</u>
Aberto (x)	<u>x</u>

As funções são definidas como (baseadas em LOGO! – SIEMENS):

- Funções Básicas (GF):

Representação no esquema de circuitos	Representação no LOGO!	Denominação da função básica
 Conexão em série Contato de trabalho		AND (E)
		AND com avaliação de flanco
 Circ. em paral. Cont. ruptura		NAND (UND não)
		NAND com avaliação de flanco
 Circuito em paralelo Contato de trabalho		OR (OU)
 Conexão em série Contato de ruptura		NOR (Ou não)
 2 Contatos do tipo inversor		XOR (exclusivo ou)
 Contato de ruptura		NOT (negação, inversor)

- Funções Especiais (SF):

Representação no esquema de circuitos	Representação no LOGO!	Denominação da função especial	Re
	Trg T	Retardamento de conexão	
	Trg R T	Retardamento de desconexão	
	Trg Par	Retard. de con. e descon.	
	Trg R T	Retardamento de conexão a ser memorizado	
	S R Par	Relais de auto-manutenção	Re
	Trg R Par	Relais de impulso de corrente	Re
	Trg T	Relais de contato passageiro	
	Trg T	Relais de contato passageiro gatilhado por flanco	
	No1 No2 No3	Interruptor de tempo para semana	
	No	Interruptor de tempo para o ano	

Representação no esquema de circuitos	Representação no LOGO!	Denominação da função especial	Re
	R Cnt Dir Par	Contador crescente e decrescente	Re
	R h Par	Contador de horas de serviço	
	En T	Gerador de impulso simétrico	
	En Inv Par	Gerador de impulso assíncrono	
	En Par	Gerador de sinal aleatório	
	Cnt Par	Interruptor de valor limiar para frequências	
	Ax Par	Interruptor de valor limiar analógico	
	Ax Ay Par	Comparador analógico	
	Trg T	Interruptor de luz de escada	
	Trg Par	Atuador confortável	
	En Nr Par	Texto de aviso	

QUESTÕES

- 1 – Defina CLP.
- 2 – O que são entradas em um CLP? Dê alguns exemplos (mínimo 3).
- 3 – O que são saídas em um CLP? Dê alguns exemplos (mínimo 3).
- 4 – Quais são as principais vantagens do PLC?
- 5 – Do que é formado ,basicamente, o hardware de um CLP?
- 6 – Defina linguagem de programação. Quais são os principais tipos de programação?
- 7 – Defina linguagem ladder? Dê algum exemplo.
- 8 – Defina linguagem de diagrama de blocos? Dê algum exemplo.
- 9 – Como funciona o endereçamento em programação de CLP (baseado em LOGO!) ?
- 10 – Qual a linguagem de programação do LOGO! (SIEMENS)?

2 – Funcionamento e Utilização do Micro C.L.P. LOGO!

O LOGO! é o módulo lógico universal da Siemens, LOGO! é um micro CLP com linguagem de programação baseada em diagrama de blocos. A Figura 2.1 apresenta o LOGO!



Figura 2.1 – Micro CLP LOGO! Siemens.

Os modelos de LOGO! disponíveis são para 12 V DC, 24 V DC, 24 V AC e 230 V AC com:

- Variante padrão com 6 entradas e 4 saídas integradas em 72 x 90 x 55mm;
- Variante sem display com 6 entradas e 4 saídas integradas em 72 x 90 x 55mm;
- Variante com 8 entradas e 4 saídas integradas em 72 x 90 x 55mm;
- Variante Long com 12 entradas e 8 saídas. Integrados em 126 x 90 x 55mm;
- Variante Bus com 12 entradas e 8 saídas, além disso, conexão para Bus com AS-Interface, através de mais 4 entradas e mais 4 saídas no sistema de Bus integrado em 126 x 90 x 55 mm.

As nomenclaturas utilizadas para diferenciar o modelo no seu nome são:

- **12:** Versão para 12 V;
- **24:** Versão para 24 V;

- **230**: Versão para 115/230 V;
- **R**: Saída de relés;
- **C**: Integrado relógio interruptor de horas;
- **o**: Variante sem display;
- **L**: número duplo de saídas e entradas;
- **B11**: Slave com conexão para Bus com AS-Interface;

A Tabela 2.1 apresenta um resumo dos modelos LOGO!

Tabela 2.1 – Modelos de LOGO!

Símbolo	Designação	Saídas	Tipo	Dimensões
	LOGO! 12/24RC *	4 x 230 V; 10A	Relais	72 x 90 x 55
	LOGO! 24 *	4 x 24 V; 0,3 A	Transistor	
	LOGO! 24RC (AC)	4 x 230 V; 10A	Relais	
	LOGO! 230RC	4 x 230 V; 10A	Relais	
	LOGO! 12/24RCo *	4 x 230 V; 10A	Relais	72 x 90 x 55
	LOGO! 24RCo (AC)	4 x 230 V; 10A	Relais	
	LOGO! 230RCo	4 x 230 V; 10A	Relais	
	LOGO! 12RCL	8 x 230 V; 10A	Relais	126 x 90 x 55
	LOGO! 24L	8 x 24 V; 0,3 A	Transistor	
	LOGO! 24RCL	8 x 230 V; 10A	Relais	
	LOGO! 230RCL	8 x 230 V; 10A	Relais	
	LOGO! 24RCLB11	8 x 230 V; 10A	Relais	126 x 90 x 55
	LOGO! 230RCLB11	8 x 230 V; 10A	Relais	

*: suplementarmente com entradas analógicas

A estrutura do LOGO! está disposta como a Figura 2.2.

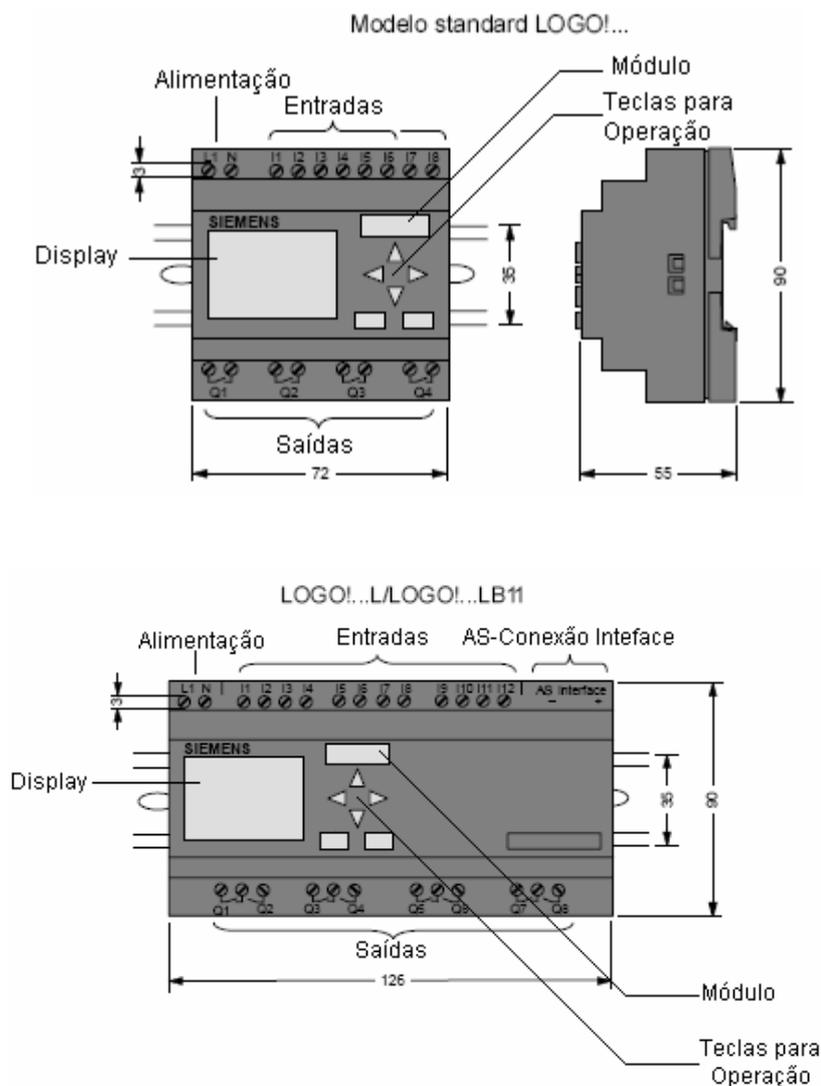


Figura 2.2 – Estrutura do LOGO!

Deve-se tomar alguns cuidados com relação à instalação do LOGO!. Tais cuidados são:

- Estar atento às normas nacionais e regionais de instalação;
- Utilizar cabos com secção entre 1,5mm² e 2,5mm²;
- Não apertar demais os bornes de conexão;
- Assentar os cabos da forma mais curta possível;
- Se necessários cabos longos, utilizar cabos blindados;
- Separar cabos CA e CC de alta tensão dos cabos de sinais de baixa tensão;
- Utilizar trilhos com perfil de 35mm de largura (DIN EM 50022);

- Não conectar uma alimentação externa paralela a uma saída DC em uma carga de saída se não utilizar diodo ou sistema de bloqueio de corrente de retorno.

Para montar o LOGO! em um painel de maneira correta deve-se:

Colocar o LOGO! no trilho com perfil e reclinar levemente o LOGO! no trilho.

A garra no lado traseiro do LOGO! deve engatar.

Para desmontar o LOGO! de maneira correta deve-se:

Introduzir uma chave de fenda no ilhós mostrado na Figura 2.3 e na parte final abaixo da garra e puxar a garra para baixo. Movimentar ligeiramente o LOGO! do trilho com perfil.

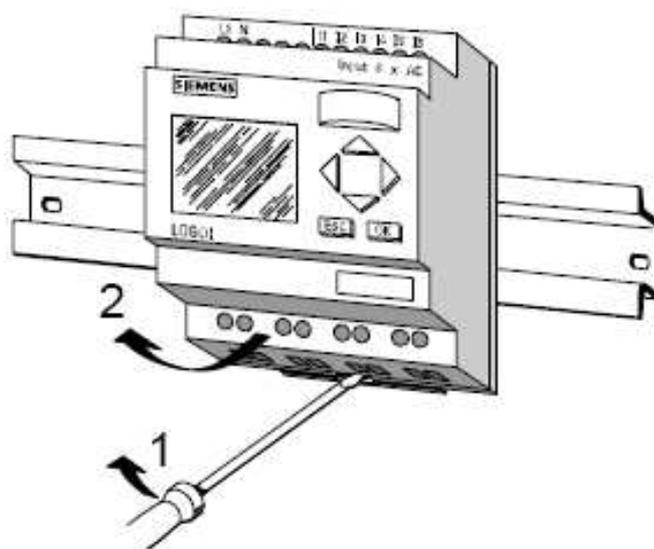


Figura 2.3 – LOGO! em um perfil de quadro elétrico.

Para conectar a alimentação de tensão ao LOGO! faça como a Figura 2.4.

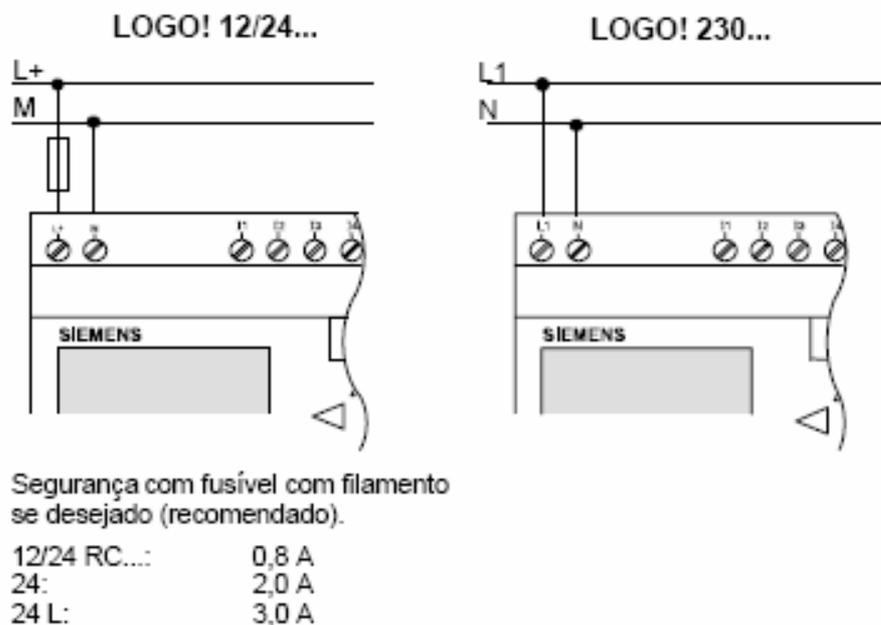


Figura 2.4 – Esquemas de alimentação para o LOGO!.

As entradas (chaves, sensores, teclas, barreiras ópticas, etc.) devem ser ligadas conforme a Figura 2.5.

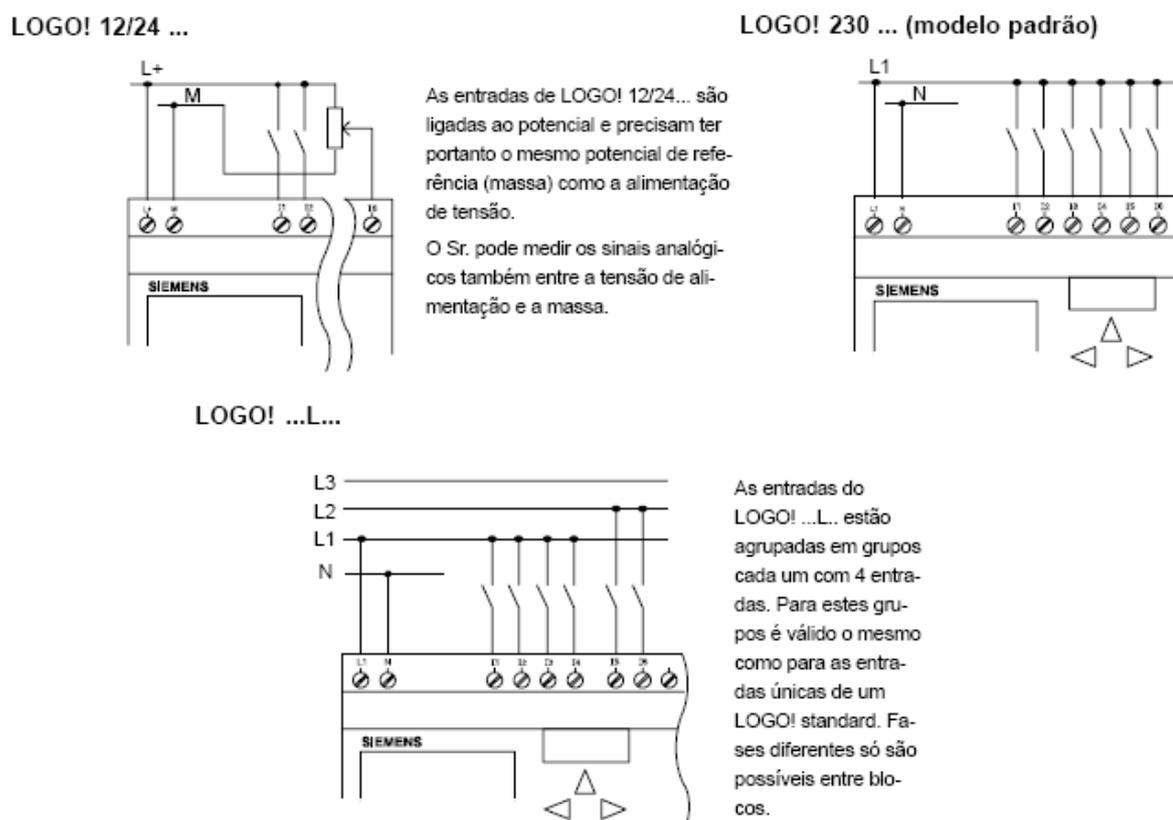


Figura 2.5 – Esquemas de ligação das entradas para o LOGO!.

Para conectar as entradas no LOGO! as mesmas devem possuir as características apresentadas na Tabela 2.2 para serem reconhecidas nos estados de conexão adequados.

Tabela 2.2 – Níveis de tensão e corrente adequados para cada estado (1 ou 0 – on/off).

	LOGO! 12/24 RC/RCo		LOGO! 24		LOGO! 24	LOGO! 230	LOGO! 12	LOGO! 24	LOGO! 24	LOGO! 230
	I1 ... I6	I7, I8	I1 ... I6	I7, I8	RC/RCo (AC)	RC/RCo	RCL	L	RCL...	RCL...
Estado de conexão 0	< 5 V DC	< 5 V DC	< 5 V DC	< 5 V DC	< 5 V AC	< 40 V AC	< 4 V DC	< 5 V DC	< 5 V DC	< 40 V AC
Corrente de entrada	< 1,0 mA	< 0,05 mA	< 1,0 mA	< 0,05 mA	< 1,0 mA	< 0,03 mA	< 0,5 mA	< 1,5 mA	< 1,5 mA	< 0,03 mA
Estado de conexão 1	> 8 V DC	> 8 V DC	> 8 V DC	> 8 V DC	> 12 V AC	> 79 V AC	> 8 V DC	> 12 V DC	> 12 V DC	> 79 V AC
Corrente de entrada	> 1,5 mA	> 0,1 mA	> 1,5 mA	> 0,1 mA	> 2,5 mA	> 0,08 mA	> 1,5 mA	> 4,5 mA	> 4,5 mA	> 0,08 mA

Durante a troca de estado deve haver um tempo mínimo para que o CLP possa reconhecer o estado solicitado. Esse tempo é o tempo de ciclo.

O tempo de ciclo pode ser calculado com o auxílio de um programa de teste conforme a Figura 2.6.

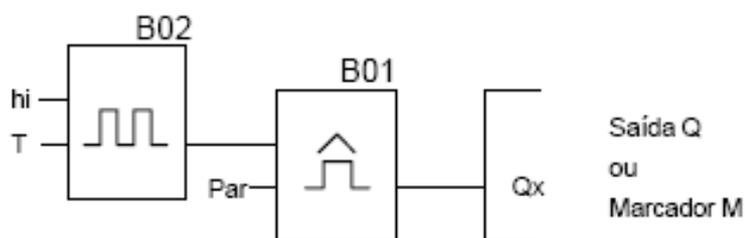


Figura 2.6 – Programa de teste para cálculo de tempo de ciclo.

E deve possuir os parâmetros conforme a Figura 2.7.

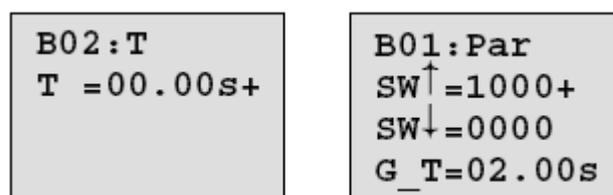


Figura 2.7 – Parametrização para o programa de teste para cálculo de tempo de ciclo.

Deve-se então partir o LOGO! e ligar o modo de parametrização e verificar o valor do bloco B01, conforme a Figura 2.8.

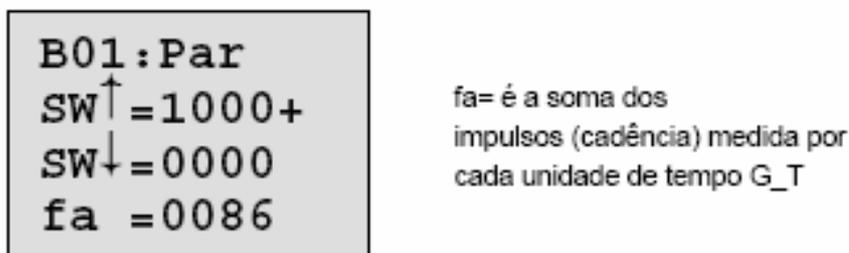


Figura 2.8 – Verificação da soma de impulsos em um ciclo de programa.

O valor recíproco de fa é o mesmo do tempo de ciclo do LOGO! com o programa atual que se encontra na memória.

$1/fa =$ tempo de ciclo em s.

LOGO! possui entradas para funções de frequências, são as entradas rápidas. As limitações de comutação de $0 \rightarrow 1$ e de $1 \rightarrow 0$ (tempo de ciclo) não são válidas para as entradas rápidas. As entradas rápidas são:

- LOGO! Versão Padrão: I5 / I6 ;
- LOGO! Versão L: I11 / I12.

Nos tipos de LOGO!24, LOGO!12/24RC e LOGO!12/24RCo as entradas I7 e I8 podem ser utilizadas como entradas analógicas utilizando as designações AI1 e AI2. Quando ligar sinais analógicos sempre utilize condutores transpostos e colocar os sensores o mais próximo possível do LOGO!.

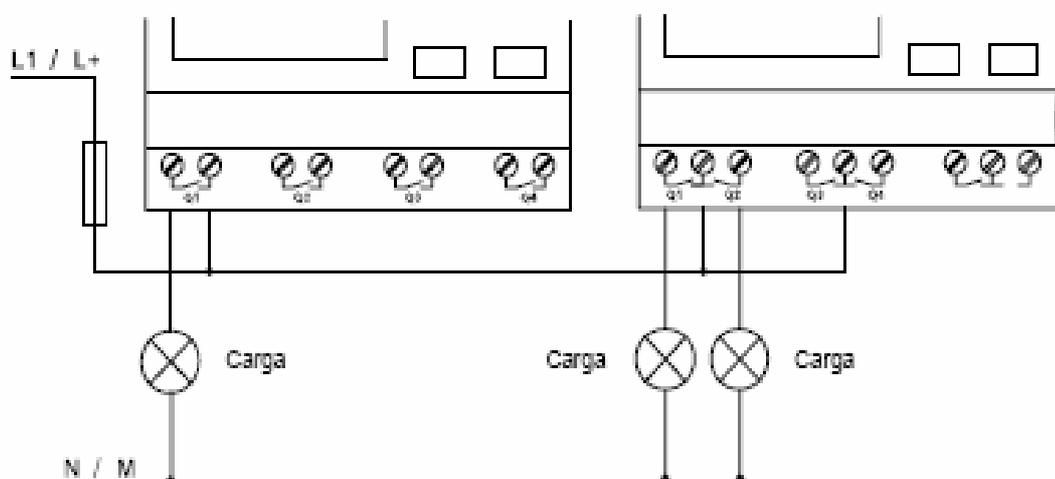
As saídas do LOGO! (R) são relés. Quanto ao potencial, os contatos dos relés são separados da alimentação de tensão e das entradas.

Nas saídas de relés podem ser conectados cargas diferentes, por ex. lâmpadas, lâmpadas fluorescentes, motores, protetores, etc. A carga conectada em um LOGO! (R) precisa apresentar as seguintes características:

- A corrente de comutação máxima depende do tipo de carga e do número desejado de manobras.

- No estado ligado ($Q = 1$) e em caso de carga ôhmica, deve ser conduzida uma corrente de no máximo 10 A (8 A tratando-se de 230 V AC), em caso de carga indutiva de no máximo 3 A (2 A tratando-se de 12/24 V AC/DC).

A Figura 2.9 apresenta o esquema de ligação para saídas em LOGO! com saída à relé.

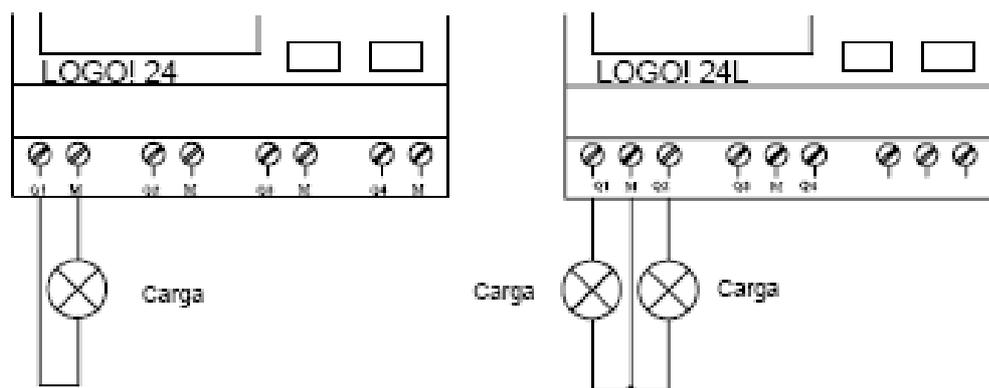


Proteção com fusível automático de segurança 16 A, característica B16, por ex.: interruptor automático 5SX2 116-6 (se desejado)

Figura 2.9 – Esquema de ligação para saídas em LOGO! com saída à relé.

Existe a variante de LOGO! com saída a transistores, pode-se reconhecer a variante LOGO! com saídas para transistores através da falta de letra da denominação do tipo **R** na nomenclatura. As saídas são resistentes a curto-circuito e resistentes à sobrecarga. Não é necessária uma alimentação separada da tensão de funcionamento visto que LOGO! suporta a alimentação de tensão da carga. Porém a corrente de comutação em cada saída deve possuir no máximo 0,3 A.

A Figura 2.10 apresenta o esquema de ligação para saídas em LOGO! com saída a transistores.



Carga: 24 V DC, 0,3 A máx.

Figura 2.10 – Esquema de ligação para saídas em LOGO! com saída a transistores.

Existe a variante LOGO!.B11 o qual se refere à comunicação entre LOGO! através de ASi-Bus.

LOGO!...B11 pode ser integrado como slave (escravo) ASi em uma rede. Através de um cabo condutor bifilar pode-se fazer a leitura e processar 4 entradas suplementares com o auxílio do ASi-Bus e operar com 4 saídas suplementares para o master (mestre) do ASi-Bus dominante.

Deve-se apenas efetuar a configuração de LOGO!...B11 no ASi-Bus com o ASi-Slave.

LOGO! ...B11 deve estar registrado no Sistema ASi, ou seja, o Busmaster atribui ao LOGO! um endereço. Mas cuidado, o endereço ASi é mutável no máximo 10 vezes para todas as variantes ...B11 do LOGO!.

Conecte o Cabo de conexão do bus no conector de tomada em anexo ou em um cabo de conexão de bus autorizado no sistema, se a polaridade estiver correta.

Introduza a tomada cablada na interface caracterizada com AS-Interface, conforme a Figura 2.11.

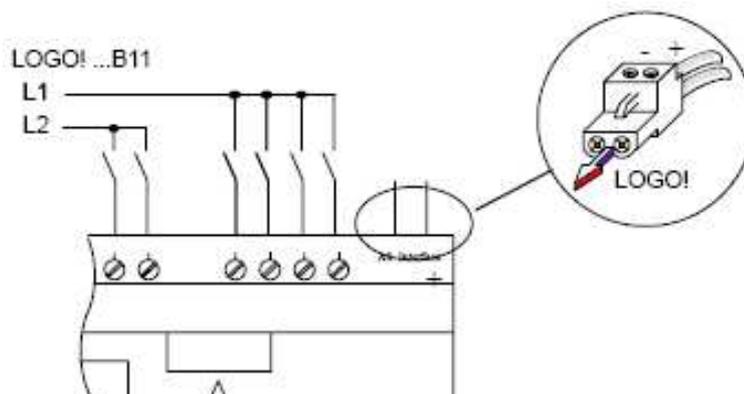


Figura 2.11 – Ligação do LOGO! ... B11 com ASi-Bus.

A fim de que se possa utilizar a função do ASi, o LOGO!...B11 precisa ser conhecido pelo Busmaster. O que ocorrerá se ligar o LOGO!...B11 com a ligação do Bus. O master ou mestre reconhece o endereço do slave ou escravo. No caso de LOGO!...B11 o endereço pré-ajustado pela fábrica = é 0. O master atribui um endereço novo desigual a 0.

Não existindo no sistema nenhum conflito a respeito do endereço ou só há um escravo com o endereço 0, não é necessário que dar outros passos.

LOGO! também possui módulos de programação, que são cartões de memória que podem arquivar programas para serem lidos posteriormente.

A localização do módulo é apresentada na Figura 1.12.

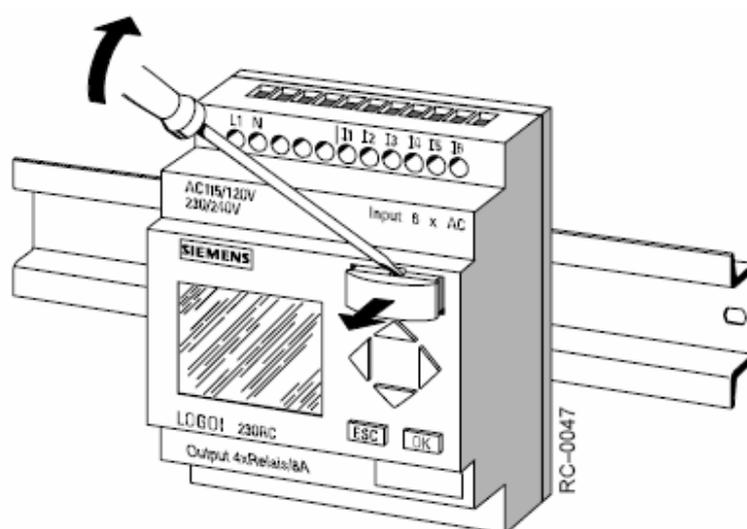


Figura 1.12 – Localização do módulo de programa em um LOGO!.

Para realizar a cópia do programa do LOGO! para o módulo de programação deve-se realizar as seguintes ações:

Pressione as teclas ←, → e OK;

Aparecerá a tela

```
>Program..
PC/Card..
Partida
```

Coloque em PC/Card e pressione OK;

Aparecerá a tela

```
>PC↔☐☐
☐☐→Card
Card→☐☐
```

Coloque em LOGO! → Card e

pressione OK;

O LOGO! copiará o programa para o módulo.

Para realizar a cópia do programa do módulo de programação para o LOGO! deve-se realizar as seguintes ações:

Pressione as teclas ←, → e OK;

Aparecerá a tela

```
>Program..
PC/Card..
Partida
```

Coloque em PC/Card e pressione OK;

Aparecerá a tela

```
>PC↔☐☐
☐☐→Card
Card→☐☐
```

Coloque em Card → LOGO! e

pressione OK;

O LOGO! copiará o programa do módulo para o LOGO!.

A capacidade (ou o espaço) de memória para funções especiais pode ser dividida em 4 zonas de memória:

- Par: Zona, na qual LOGO! memoriza os valores nominais.

- RAM: Zona na qual LOGO! deposita os valores reais.
- Timer: Zona que o LOGO! utiliza para as funções de tempo.
- REM: Zona na qual LOGO! deposita valores atuais e reais a serem mantidos.

Um programa em LOGO! pode ocupar no máximo os recursos apresentados na Tabela 2.3. E a Tabela 2.4 apresenta os recursos de memória de cada função especial.

Tabela 2.3 – Recursos de memória disponíveis.

Blocos	Par	RAM	Timer	REM	Marcador
56	48	27	16	15	8

Tabela 2.4 – Recursos de memória de cada função.

Bloco de funções	Par	RAM	Timer	REM
Relais de automanutenção*	0	(1)	0	(1)
Relais de impulso de corrente*	0	(1)	0	(1)
Relais de contato passageiro	1	1	1	0
Relais de contato passageiro gatilhado por flanco	1	1	1	0
Retardamento de conexão	1	1	1	0
Retardamento de desconexão	2	1	1	0
Retardamento de conexão e de desconexão	2	1	1	0
Retardamento de conexão a ser memorizado	2	1	1	0
Interruptor de tempo para semana	6	2	0	0
Interruptor de tempo para o ano	2	0	0	0
Contador crescente e decrescente*	2	(2)	0	(2)
Contador de horas de serviço	2	0	0	4
Gerador de impulsos simétrico	1	1	1	0
Gerador de impulso assíncrono	3	1	1	0
Gerador de sinal aleatório	2	1	1	0
Interruptor de valor limiar para frequências	3	3	1	0
Interruptor de valor limiar analógico	4	2	0	0
Comparador analógico	3	4	0	0
Interruptor de luz de escada	1	1	1	0
Atuador confortável	2	1	1	0
Texto de aviso	1	0	0	0

Um caminho de programa é composto de uma série de blocos de funções, que por sua vez começam e terminam com um bloco terminal. O número de blocos em um caminho de programa descreve o nível de embutimento.

Blocos terminais são entradas e nível (I, Ia, Hi, Lo), bem como saídas e marcadores (flags) (Q, Qa, M).

Programar é transformar aplicação/circuito em programas de linguagem do LOGO!

Para realizar a programação, deve-se ter conhecimento dos conceitos:

- Borne;
- Bloco.

BORNE (Co) refere-se às conexões e estados disponíveis no LOGO!.

A Tabela 2.5 apresenta os tipos de borne.

Tabela 2.5 – Tipos de bornes

Bornes			
Entradas	I1... I6, I7 (AI1), I8 (AI2)	I1...I12	I1...I12 e Ia1...Ia4 (AS-Interface)
Saídas	Q1...Q4	Q1...Q8	Q1...Q8 e Qa1...Qa4 (AS-Interface)
lo	Sinal com nível '0' (desl)		
hi	Sinal com nível '1' (lig)		
x	Não é utilizada uma conexão existente		

BLOCO (BN), representando funções (GF – Funções básicas e SF – Funções especiais) as quais transformam as informações de entrada em informações de saída. A Figura 2.13 apresenta a estrutura do bloco.

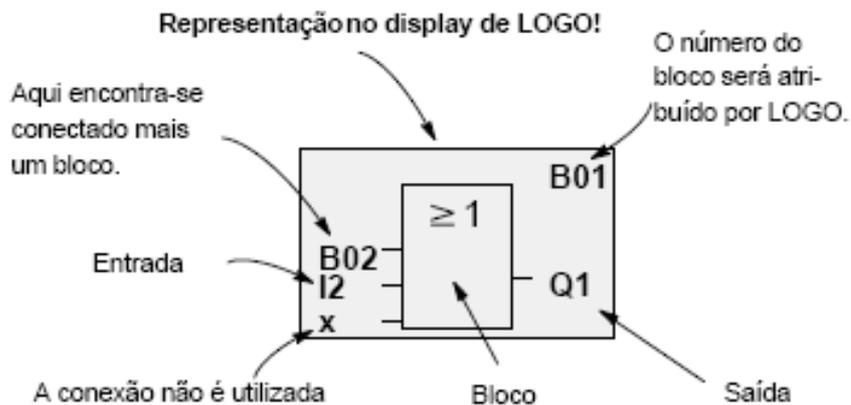
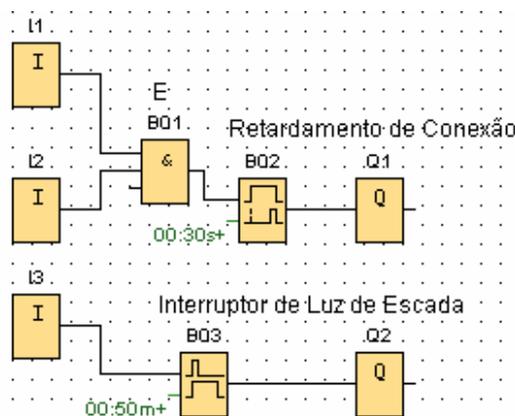


Figura 2.13 – Estrutura de bloco.

O número do bloco será introduzido assim que o bloco for inserido. O número é importante para interligação entre blocos e facilidade em parametrizar sistemas.

QUESTÕES

- 1 – O que é LOGO!?
- 2 – Quais são os cuidados com a instalação do LOGO!?
- 3 – Desenhe as formas de alimentação do LOGO!.
- 4 – Desenhe as formas de conectar as entradas no LOGO!.
- 5 – Desenhe as formas de conectar as saídas no LOGO!.
- 6 – Faça o cálculo de memória e o nível de embutimento para o seguinte programa.



- 7 – O que é Borne? O que é Bloco?

3 - LABORATÓRIO – PROGRAMAÇÃO MANUAL

Vamos agora realizar uma programação manual do esquema representado pela Figura 3.1 para a fixação.

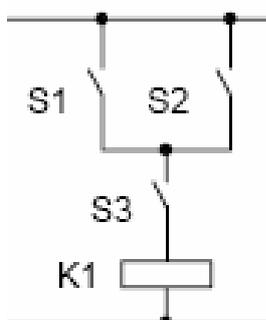


Figura 3.1 – Esquema de ligação para exemplificação.

O sistema deverá ser concluído conforme a Figura 3.2.

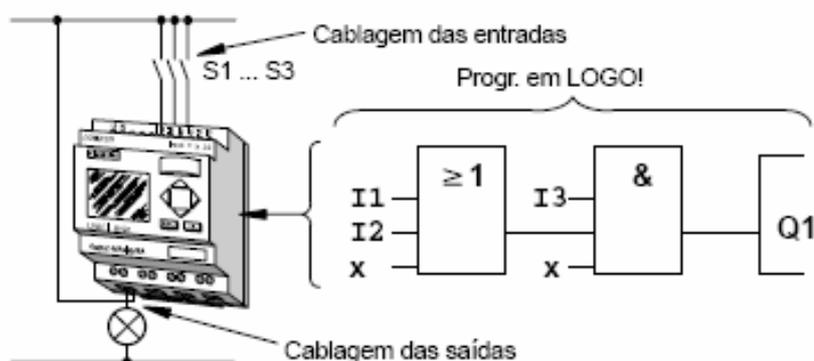
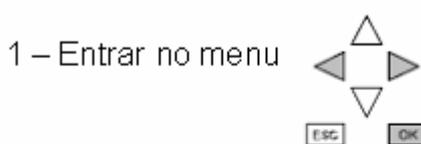
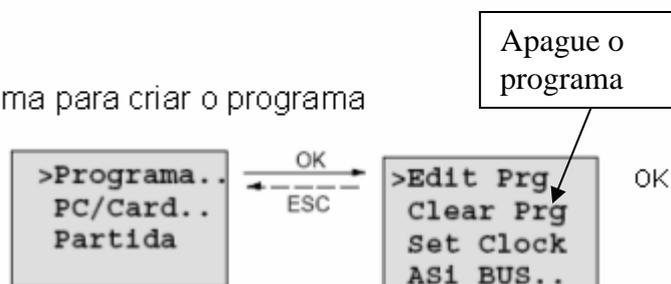


Figura 3.2 – Sistema em LOGO! do exemplo proposto.

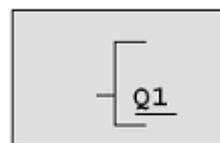
Primeiramente energize a fonte de alimentação do LOGO!. Os passos a seguir deverão ser realizados para a montagem completa do sistema. Em LOGO! deve-se iniciar pela **saída do programa** e ir voltando até o **início**. Por isso é necessário desenhar o programa primeiramente.



2 – Entrar no modo editar programa para criar o programa

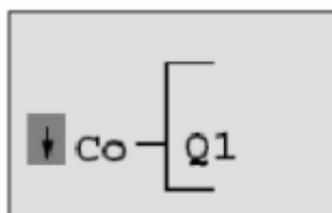


3 – O LOGO! mostrará apenas a primeira saída com as teclas para cima e para baixo é possível selecionar as outras saídas.



4 – Com a tecla ← posicione o cursor para inserir o bloco B01 AND para as entradas I3 e a saída do bloco B02.

5 – Pressione OK



O cursor está representado por um quadrado preto: pode selecionar um borne ou um bloco

6 – Pressionando para baixo, selecione GF e pressione OK, o primeiro bloco da relação de funções básicas (AND) aparecerá. Pressione OK



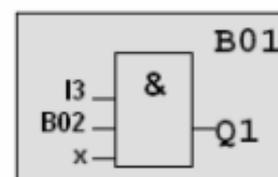
7 – Pressione OK novamente para selecionar as entradas para o bloco. Selecione (Co) para a primeira entrada e pressione OK. Selecione a Entrada I3 e pressione OK.

Para a segunda entrada selecione (BN) e pressione OK.

Selecione o bloco B02 e pressione OK.

Para a terceira entrada selecione (Co) e pressione OK.

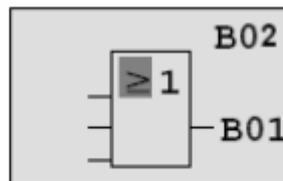
Selecione a entrada x e pressione OK.



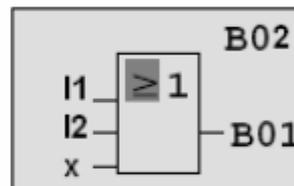
8 – Com a tecla ← posicione o cursor para inserir o bloco **B02** OR para as entradas **I1** e **I2**. E pressione OK.

Selecione (GF) e pressione OK.

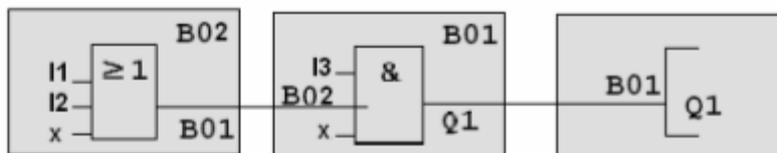
Selecione o bloco da função OR e pressione OK.



9 – Conforme exemplificado anteriormente, insira **I1**, **I2** e **x** nas respectivas entradas e pressione OK ao final.



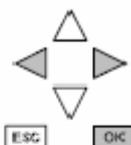
Finalmente o programa completo:



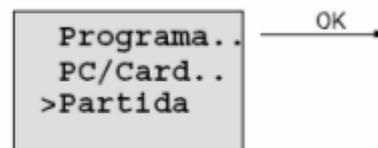
Pressione ESC.

Com relação à partida no equipamento (colocar em modo RUN) deve-se:

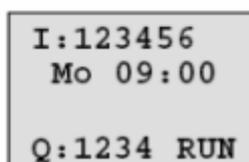
1 – Entrar no menu



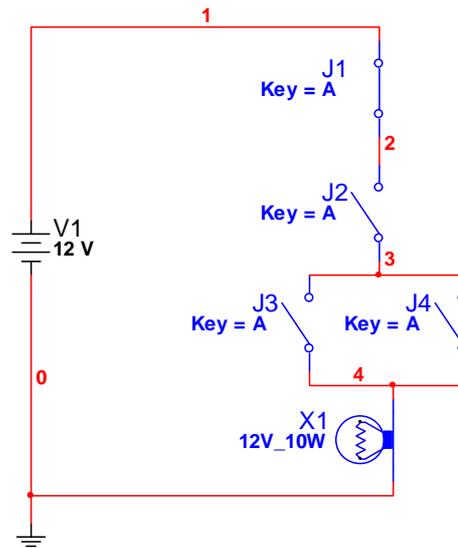
2 – Entrar no modo Partida pressionando para baixo e OK



3 – O LOGO! Entrará no modo RUN.



Agora faça o programa para o circuito abaixo:



BIBLIOGRAFIA

CAPELLI, Alexandre. *CLP Controladores Lógicos Programáveis na Prática*. Rio de Janeiro: Antenna Edições Técnicas Ltda, 2007.

GEORGINI, Marcelo. *Automação Aplicada – Descrição e Implementação de Sistemas Seqüenciais com PLCs*. 7 ed. São Paulo: Editora Érica, 2006.

Manual de Instruções do LOGO! (SIEMENS). A5E00067788 01.

Manual do Curso STEP 7 – SIEMENS (BASICS OF PLC).

ORTEGA, José. *Curso de Programación y mantenimiento de PLC's industriales*.

PARR, E. A. *Programmable Controllers An Engineering's Guide*. 3 Ed. Amsterdam: Newnes, 2003.