

Parte 03 - Linguagens para programação de CLP (Norma IEC 61131 – 3)

Introdução:

Há muito tempo se procura estabelecer um padrão para programação de CLP, em 1979, foi designado um grupo de trabalho com o IEC (International Electro-technical Commission) voltado para este propósito.

Este grupo tinha como objetivo analisar o projeto completo de CLP's (inclusive hardware), instalação, testes, documentação, programação e comunicações. Este grupo designou oito frentes de trabalho para desenvolver diferentes partes do padrão para CLP's.

Em 1992, o IEC publicou a norma IEC 1131, a qual estabelece padrões para Controladores Lógicos Programáveis. Em suas diversas versões a norma ganhou o número 6 passando assim, para IEC 61131. Esta está dividida em partes, que são:

- ◇ 61131-1 - Informações gerais
- ◇ 61131-2 - Requisitos de hardware
- ◇ 61131-3 - Linguagens de programação
- ◇ 61131-4 - Guia de orientação ao usuário
- ◇ 61131-5 – Comunicação

Outras três partes, ainda em fase de elaboração.

- ◇ 61131-6 - Comunicação via *Fieldbus*
- ◇ 61131-7 - Programação utilizando Lógica *Fuzzy*
- ◇ 61131-8 - Guia para implementação das linguagens

Em 1993, é publicada sua terceira parte a IEC 61131-3 estabelecendo um padrão global para programação de controladores lógicos programáveis. Nasce assim, uma interface padrão permitindo que pessoas com diferentes habilidades e formações, criem programas durante estágios diferentes do ciclo de vida de um software. Fazem parte deste ciclo: especificação, projeto, implementação, teste, instalação e manutenção.

3.1 IEC 61131- 3 (Linguagens de programação)

A adoção da IEC 61131-3 pelos diversos fabricantes de sistemas de controle é uma realidade inegável. Assim, todo profissional da área, seja técnico ou engenheiro deve conhecê-la.

Hoje, a IEC 61131-3 é o único padrão global para programação de controle industrial que consiste na definição da linguagem que é a **Função gráfica de seqüenciamento (SFC)**, usada para estruturar a organização interna do programa, e de quatro linguagens, sendo duas textuais: **Lista de Instrução (IL)** e **Texto Estruturado (ST)** e duas gráficas: **Diagrama de blocos de funções (FBD)** e **Diagrama Ladder (LD)**.

Cabe ao projetista/programador escolher a linguagem que melhor se adapta ao seu sistema, daí a necessidade de conhecer uma pouco de cada uma, não faz parte do escopo deste trabalho detalhar cada uma delas e sim cita-las descrevendo suas características e apresentando um modelo.

3.1.1 Sequential Function Chart (SFC):

São gráficos de função seqüencial, originou-se na França e teve como base a redes de petri e o Grafcet (*Grphe Fonctionnel de Command Etape Transition*), em 1988 foi publicado tornando-se padrão internacional.

Muito mais que uma linguagem o SFC descreve o comportamento do programa, seja ele, seqüencial paralelo ou misto, além de organizar a sua estrutura interna, ajudando a decompor um problema de controle em partes gerenciáveis, enquanto mantém uma visão global da solução do problema.

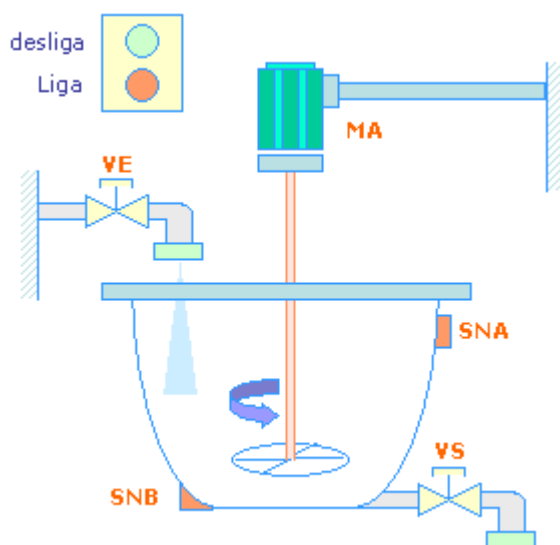
Principais características

- ◇ É usada na estruturação do programa, não importando a linguagem utilizada.
- ◇ Fácil representação e interpretação
- ◇ Facilidade de diagnóstico (localização de falhas)
- ◇ Permite gerar divergências e convergências de seqüências.
- ◇ Descreve o comportamento do sistema através de passo transições e ações. Sendo:
 - Passo: estado do programa onde as ações são executadas.
 - Transição: condição pela qual o programa muda de estado, passando de um ou mais passos antecedentes para um ou mais passos sucessores.
 - Ação: atividade de controle executada num determinado passo.

3.1.1.1 – Implementação prática em SFC - Tanque agitador

Deseja-se implementar um sistema de controle para um tanque misturador simples, como mostrado no esquema:

Ilustração do sistema



Definindo I/O

Entradas

BL = Botão de liga
 BD = Botão de desliga
 SNA = Sensor nível alto
 SNB = Sensor nível baixo

Saídas

VE = válvula de entrada
 MA = Motor de Agito
 VS = válvula de saída

Figura 3.1 – Tanque agitador

Descrição de funcionamento:

Ao pressionar o botão de liga (BL) a válvula de entrada (VE) é acionada e o tanque começa a encher. Quando o sensor de nível alto (SNA) for atingido, a válvula de entrada (VE) é fechada ligando o motor de agito (MA) que permanece ligado por 10 segundos. Em seguida a válvula de saída (VS) é ligada, quando o sensor de nível baixo (SNB) for acionado o ciclo recomeça. Se o botão de desliga (BD) não for pressionado o ciclo recomeça.

A estrutura do SFC para o tanque agitador é mostrada na figura 3.2.

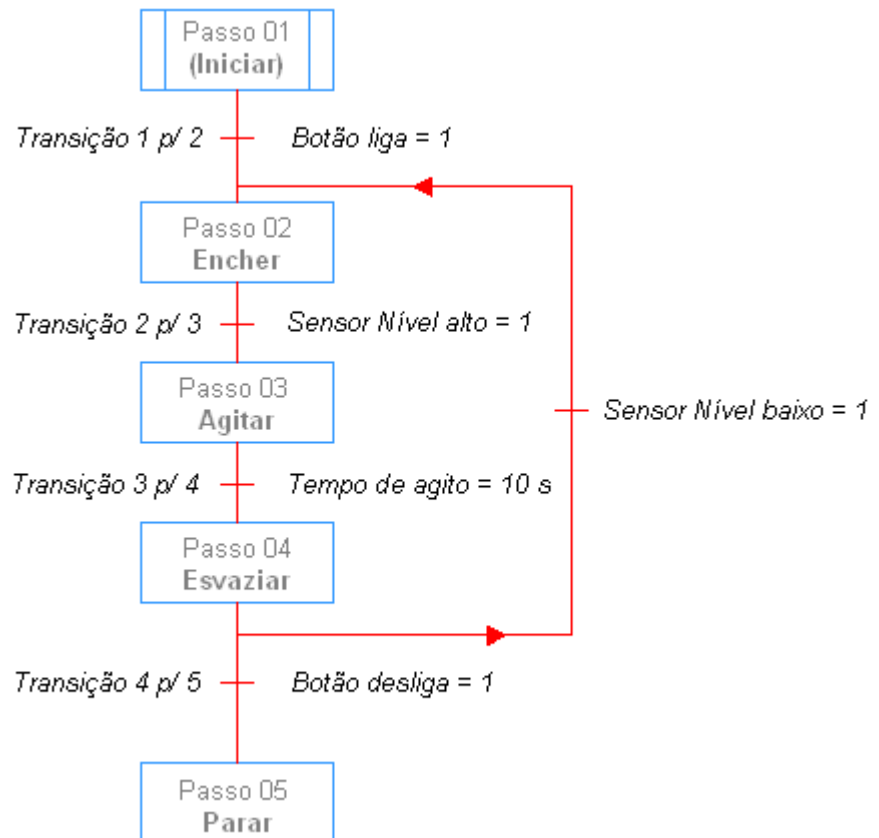


Figura 3.2 - A estrutura do programa SFC

3.1.2 Instruction List (IL)

É uma linguagem textual, próxima do código de máquina, é ideal para resolver problemas simples onde existem poucas quebras no fluxo de execução. Na verdade é apenas uma linguagem adicional, menos amigável e flexível e que deve ser usada para produzir código otimizado para trechos de performance crítica em um programa.

Principais características:

- ◇ Linguagem de Baixo Nível
- ◇ Semelhante ao *Assembler*
- ◇ Ideal para pequenas aplicações ou otimização de códigos
- ◇ Linguagem básica para exportação de programas (Portabilidade)

3.1.2.1 Implementação prática em IL – Função ou exclusivo

Deseja-se implementar uma função OU exclusivo, ou seja, fornece 1 (um) à saída quando as variáveis de entrada forem diferentes entre si.

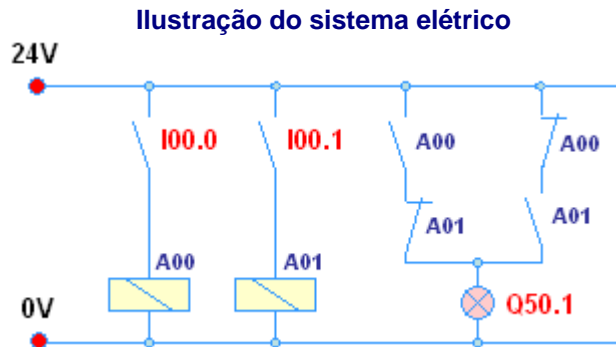


Tabela verdade		
I00.0	I00.1	Q50.0
0	0	0
0	1	1
1	0	1
1	1	0

Figura 3.3 – OU exclusivo elétrico

A lista de instrução para a função OU exclusivo

LD	I00.0	* carrega a entrada I00.0
ANDN	I00.1	* faz um and lógico entre I00.0 e I00.1 invertido
OR		
(
LDN	I00.0	* carrega a entrada I00.0 invertida
AND	I00.1	* faz um and lógico entre I00.0 invertido e I00.1
)		* faz o OU lógico entre as duas expressões
ST	Q50	* carrega a saída Q50.0

Figura 3.4 – Estrutura do programa IL

3.1.3 Structured Text (ST)

Também é uma linguagem textual, porém de alto nível, que permite a programação estruturada. A vantagem do texto estruturado esta na utilização de sub-rotinas para executar diferentes partes de uma função de controle.

Principais características:

- ◇ Linguagem de alto nível
- ◇ Semelhante ao Pascal (ISO 7185)
- ◇ Ideal para:
 - Tomada de decisões
 - Declarações (Variáveis, POUs, Configurações, etc.)
 - Cálculos
 - Implementação de algoritmos
 - Definição de ações (SFC)
 - Utilização de literais
 - Criação de blocos Etc.

3.1.2.1 Implementação prática em ST – Liga/ desliga motor

O motor (**M**) ficará energizado se, e somente se, o botão liga (**I1**) for acionado e o botão desliga (**I0**) não for acionado. Quando o motor (**M**) estiver energizado, o indicador luminoso (**L**) também estará energizado. (**M**) e (**L**) ficarão desenergizadas caso o botão desliga (**I0**) seja acionado.

O esquema elétrico

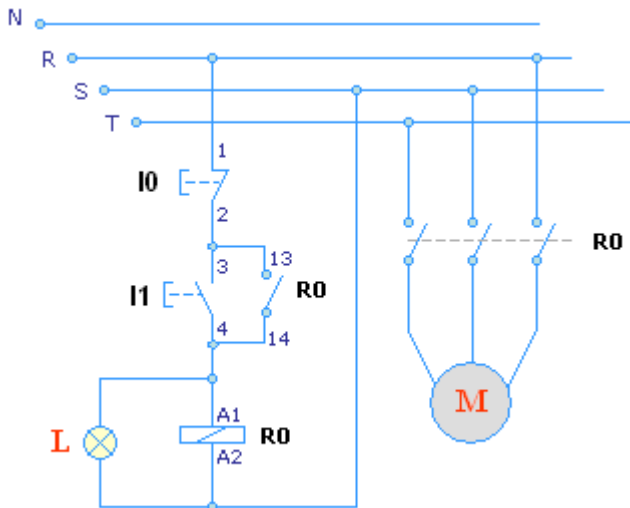


Figura 3.5 – Partida direta de motores

Texto estruturado

```

IF I1
OR M AND N IO
THEN SET M
IF M
THEN SET L
OTHRW RESET M
    
```

Figura 3.6– Estrutura do programa ST

3.1.4 Function Block Diagram (FBD)

É uma linguagem gráfica, e por isso é muito mais amigável que as textuais é baseada nos circuitos lógicos, portanto muito semelhante as portas lógicas estudadas em eletrônica digital. Também pode ser usado para modelar sistema em termos do fluxo de sinais entre elementos de processamento.

Principais características:

- ◇ Adequada para controle discreto, seqüencial, regulatório, etc.
- ◇ Representação de fácil interpretação
- ◇ Blocos expansíveis em função do no de parâmetros de entrada
- ◇ São disparados por parâmetros externos, enquanto os algoritmos internos permanecem escondidos.
- ◇ Blocos encapsulam o algoritmo, destacando o fluxo de informações e o processamento de sinais.

3.1.4.1 Implementação prática em FBD – segurança em prensas

Duas chaves devem comandar uma prensa simultaneamente de modo que acionada a primeira chave, não podem transcorrer mais do que 0,5s até que a segunda chave seja acionada. Se o operador retirar a mão das chaves, a prensa deverá parar, por razões de segurança.

Ilustração

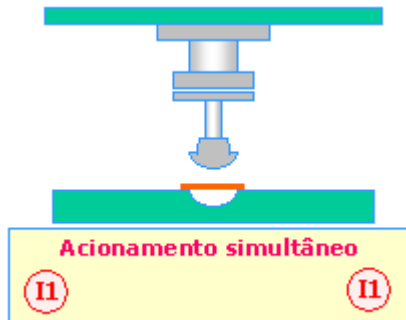


Figura 3.7 – Ilustração da máquina prensa peças

Diagrama de blocos funcionais

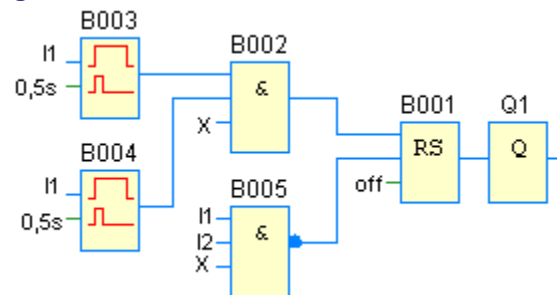


Figura 3.8– Estrutura do programa FBD

3.1.5 Ladder Diagram (LD)

É uma linguagem gráfica, muito amigável, foi baseada na lógica de contatos o que a torna de fácil compreensão no meio elétrico. É a linguagem foco deste trabalho, assim no próximo a linguagem ladder será detalhada.

Principais características:

- ◇ Baseada no diagrama elétrico de contatos
- ◇ Adequada para controle discreto, combinacional e seqüencial
- ◇ Utilizam blocos de função para controle regulatório e funções especiais.

3.1.5.1 Implementação prática em LD – Partida direta reversa

Deseja-se implementar em ladder uma partida direta reversa de motores trifásicos, que consiste em mudar o sentido de rotação de um motor trifásico. Sua seqüência operacional é bastante simples:

Pressionando (S1) energiza-se o contator (K1), fechando o seu selo (13,14) e abrindo o intertravamento (21,22) mesmo pressionando (S2) o contator (K2) não será energizado, devido ao intertravamento, sendo necessário seu desligamento para religar (S2) novamente e a rotação será contrário.

O esquema elétrico da partida direta reversa

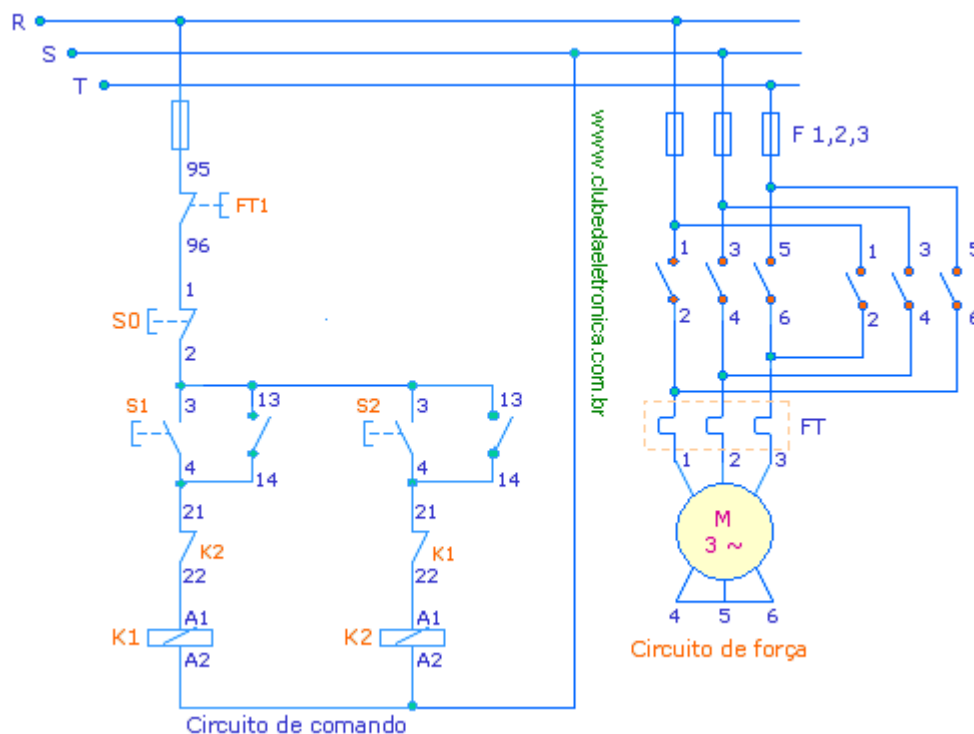


Figura 3.9– Partida direta e reversa

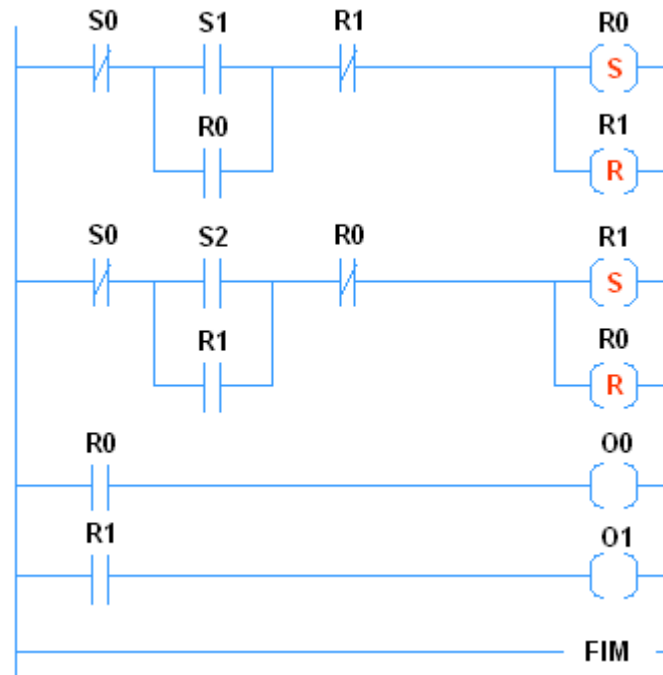
Ladder correspondente

Figura 3.10– Estrutura do programa LD

Dentre as linguagens descritas, daremos um enfoque ao diagrama ladder, a razão é simples é a mais utilizada na indústria. Assim, será trabalhada no próximo capítulo.

***Há duas formas de enfrentar dificuldades: alterá-las ou alterar sua maneira de enfrentá-las.
Phyllis Bottome.***

www.clubedaeletronica.com.br

Referências bibliográficas:

- ❑ http://www.plcopen.org/pages/tc1_standards/iec_1131_or_61131/
- ❑ <http://www.cpdee.ufmg.br/~carmela/NORMA%20IEC%201131.doc>
- ❑ <http://www.software.rockwell.com/corporate/reference/iec1131/>
- ❑ <http://www.plcopen.org/>
- ❑ <http://www.lme.usp.br/~fonseca/psi2562%20aula%206%20IHM.pdf>
- ❑ <http://www.teses.usp.br/teses/disponiveis/18/18133/tde-11072002-085859/>
- ❑ http://www.redenet.edu.br/publicacoes/arquivos/20080108_144615_INDU-058.pdf
- ❑ <http://www.corradi.junior.nom.br/modCLP.pdf>
- ❑ <http://www.cpdee.ufmg.br/~seixas/Paginall/Download/DownloadFiles/>