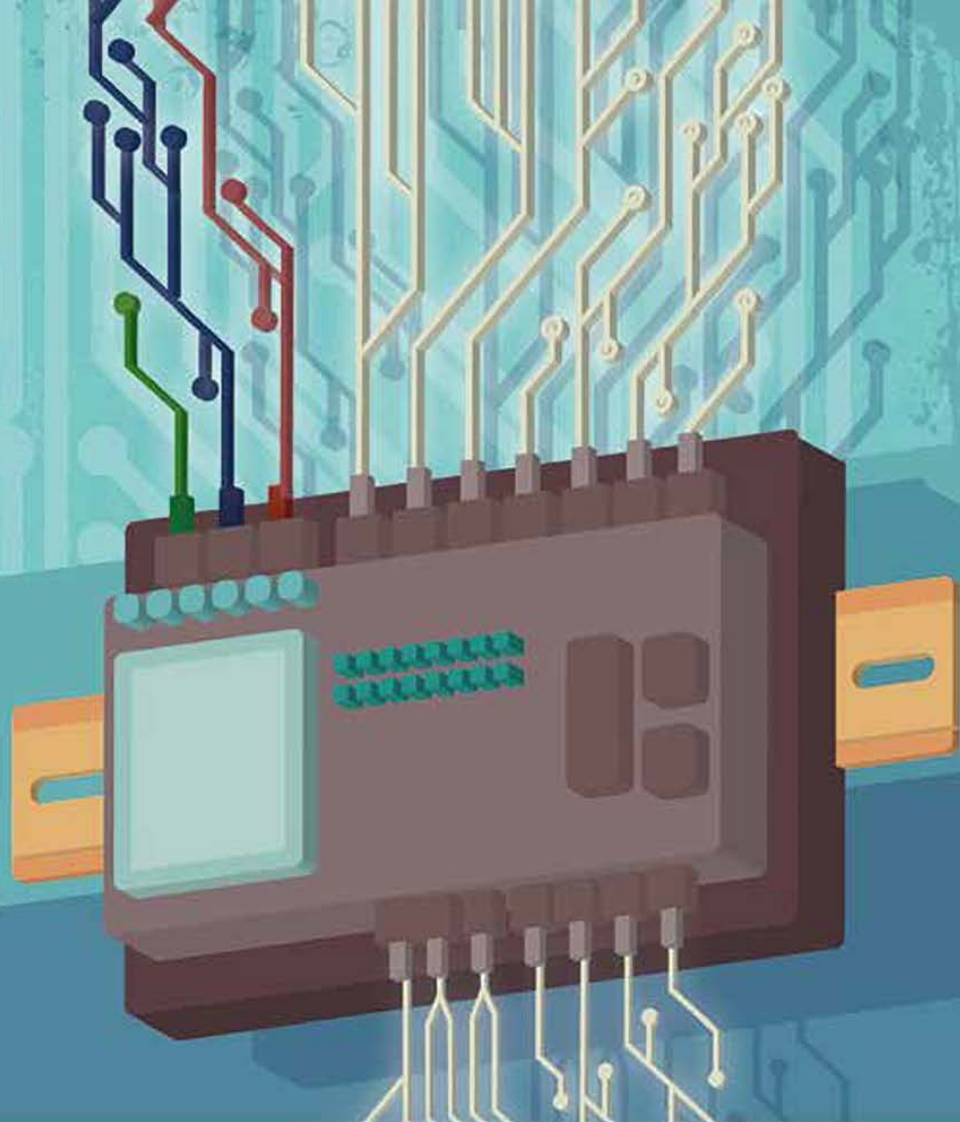




Wânderson de Oliveira Assis
Alessandra Dutra Coelho

AUTOMAÇÃO INDUSTRIAL UTILIZANDO PLCs

(PROGRAMMABLE LOGIC CONTROLLERS)

WÂNDERSON DE OLIVEIRA ASSIS
ALESSANDRA DUTRA COELHO

AUTOMAÇÃO INDUSTRIAL UTILIZANDO PLCs
(PROGRAMMABLE LOGIC CONTROLLERS)

Pós-graduação em Engenharia de Automação e Controle Industrial do
Centro Universitário do Instituto Mauá de Tecnologia.

SÃO CAETANO DO SUL
2025



Reitor

Prof. Dr. Marcello Nitz da Costa

Pró-Reitora de Graduação

Profa. Dra. Ana Paula Gonçalves Serra

Pró-Reitor Acadêmico

Prof. Dr. Fernando Silveira Madani



Equipe editorial

Prof. Dr. Everaldo Pereira

Prof. Dr. João de Sá Brasil Lima

Prof. Dr. Octavio Mattasoglio Neto

Prof. Dr. Rodrigo Fernando Gallo

Profa. Dra. Vanessa Seriacopi

Bibl. Felipe Augusto S. dos S. Rio

Branco

Atribuição-Não Comercial-Sem
Derivações 4.0 Internacional
Esta licença permite o download e
compartilhamento da obra desde que
sejam atribuídos créditos ao(s)
autor(es), sem a possibilidade de
alterá-los ou utilizá-la para fins
comerciais.

Capa

Livia Cassimiro Farias

Dados Internacionais de Catalogação na Publicação (CIP) Biblioteca Engenheiro Álvaro de Souza Lima - IMT

A848a Assis, Wânderson de Oliveira Assis.
Automação Industrial Utilizando PLCs (Programmable Logic
Controllers) [recurso eletrônico] / Wânderson de Oliveira Assis
, Alessandra Dutra Coelho. Dados eletrônicos. -- São Caetano
do Sul: PLAM, 2025.
1 recurso online (151 p. : il., PDF)

Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-65-83488-02-2

1. Automação Industrial. 2. Dispositivos lógicos
programáveis. 3. Sistemas de tempo discreto. 4. Controladores
programáveis. 5. Controladores PID. I. Coelho, Alessandra
Dutra. II. Título.

CDU 621.3.049.77

Elaborada por Felipe Augusto Souza dos Santos Rio Branco – CRB8/9104

Publicação digital - 1. edição - agosto - 2025



PLAM PORTAL DE LIVROS
ABERTOS DA MAUÁ

Praça Mauá, 1 – CEP 09580-900 - São Caetano do Sul – SP
plam.maua.br

AGRADECIMENTOS

Gostaria de agradecer a todos aqueles que contribuíram para a execução deste trabalho.

Agradeço também ao suporte oferecido pelo corpo técnico das Engenharias Elétrica e Eletrônica do Instituto Mauá de Tecnologia.

Por último, e o mais importante, agradeço a Deus por nos dado paciência e persistência para concluir este trabalho.

RESUMO

Este trabalho apresenta os diversos conceitos associados ao desenvolvimento de aplicações com controladores lógicos programáveis (*programmable logic controllers* – PLCs). A abordagem apresenta os principais fundamentos necessários para o projeto de sistemas de controle em eventos discretos incluindo, conceituação de sistemas a eventos discretos, apresentação dos principais conceitos relativos à norma IEC 61131, fundamentos da programação em diagrama de contatos (*ladder*) e SFC (*Sequential Flow Chart*) e Lista de Instruções (*Statement List*). Como referência são apresentados exemplos em PLCs fabricados pelas empresas Omron, Schneider Electric e Festo. O trabalho apresenta também os conceitos básicos de pneumática e o desenvolvimento de aplicações de pneumática controladas por PLCs. São também demonstrados conceitos de redes industriais e sistemas supervisórios além de noções de controle de processos utilizando algoritmos PID desenvolvidos em plataformas com controladores lógicos programáveis.

Palavras-chave: Controle de Processos em Eventos Discretos, Automação Industrial, Controladores Lógicos Programáveis, Pneumática, Redes de Automação e Sistemas Supervisórios, Controle PID.

ABSTRACT

This work presents several concepts associated with the development of applications with programmable logic controllers (PLCs). The approach presents the main basic fundamentals related to design of discrete events control systems, including concepts of discrete event systems, the basic of standard IEC 61131, fundamentals of programming in ladder, SFC (Sequential Flow Chart) and Statement List. As reference are presented examples using Omron, Schneider Electric and Festo PLCs. The work also present basic concepts of pneumatic and the development of pneumatic applications controlled by PLCs. Concepts of industrial networks and supervisory systems are also demonstrated, as well as notions of process control using PID algorithms developed on platforms with programmable logic controllers.

Palavras-chave: Discrete Event Process Control, Industrial Automation, Programmable Logic Controllers, Pneumatic, Automation Networks and Supervisory Systems, PID Control.

LISTA DE FIGURAS

FIGURA 1.1	Automação rígida e automação flexível.....	1
FIGURA 1.2	Instrumentos para medição de variáveis contínuas.....	2
FIGURA 1.3	Exemplos de sistemas utilizados no controle de SVCs	3
FIGURA 1.4	Central de Controle de um SDCD.....	3
FIGURA 1.5	Aplicações de sistemas a eventos discretos tipicamente controlados por CLPs	5
FIGURA 1.6	Outras aplicações de sistemas a eventos discretos	6
FIGURA 1.7	Controle em batelada	7
FIGURA 2.1	Diagrama de blocos de um CLP	11
FIGURA 2.2	Ciclo de varredura	12
FIGURA 2.3	Alguns modelos de controladores lógicos programáveis	13
FIGURA 2.4	Alguns modelos de controladores lógicos programáveis Omron.....	14
FIGURA 2.5	Expansão de I/O's utilizando sistema de expansão de racks.....	17
FIGURA 2.6	Entradas digitais	18
FIGURA 2.7	Saídas digitais	18
FIGURA 2.8	Entradas e saídas analógicas	19
FIGURA 2.9	Entrada DC - Circuito de acoplamento ótico para entradas digitais - CLP Omron	19
FIGURA 2.10	Entrada AC - Circuito de acoplamento ótico para entradas digitais - CLP Omron	20
FIGURA 2.11	Saída digital a transistor PNP - Circuito de acoplamento ótico - CLP Omron	20
FIGURA 2.12	Saída digital a Triac - Circuito de acoplamento ótico - CLP Omron.....	21
FIGURA 2.13	Saída digital a relé - Circuito de acoplamento ótico - CLP Omron.....	21
FIGURA 2.14	Exemplo de progr. utilizando as linguagens previstas Norma IEC-61131	27
FIGURA 2.15	Utilização do Mapa de Karnaugh na obtenção do programa em ladder	28
FIGURA 2.16	Fluxograma	29
FIGURA 2.17	Variáveis de estado	30
FIGURA 2.18	Diagrama trajeto-passo	30
FIGURA 3.1	Comandos básicos da programação em ladder	32
FIGURA 3.2	Programas em ladder para representação de funções lógicas básicas.....	34

FIGURA 3.3	Varredura em duas lógicas consecutivas	35
FIGURA 3.4	Uma aplicação constituída somente de uma lógica	36
FIGURA 3.5	Uma aplicação constituída de duas lógicas separadas	37
FIGURA 3.6	Circuito de intertravamento	38
FIGURA 3.7	Utilização de bobinas retentivas SET e RESET	38
FIGURA 3.8	Circuito de detecção de borda	39
FIGURA 3.9	Formas de onda obtidas no circuito de detecção de borda	40
FIGURA 3.10	Uma aplicação utilizando entrada pulsante	41
FIGURA 3.11	Uma aplicação com circuito de detecção de borda controlado por bobinas retent	42
FIGURA 3.12	Estrutura básica de linha de programação de um temporizador em ladder	43
FIGURA 3.13	Formas de onda em uma aplicação com temporizador	44
FIGURA 3.14	Uma aplicação com temporizador	45
FIGURA 3.15	Oscilador astável construído com temporizadores	45
FIGURA 3.16	Uma aplicação básica com temporiz. TOFF e formas de onda resultantes	46
FIGURA 3.17	Uma aplicação básica com temporizador	47
FIGURA 4.1	CLP Modular Omron incluindo módulos I/O e módulos de rede ...	50
FIGURA 4.2	CLP Modular Omron conectado a módulo remoto DeviceNet	50
FIGURA 4.3	Janela de inicialização do programa CX - Programmer	52
FIGURA 4.4	Seleção do CLP que será utilizado	52
FIGURA 4.5	Criação do I/O Table	53
FIGURA 4.6	Modos de operação	54
FIGURA 4.7	Paleta de programação em Ladder	57
FIGURA 4.8	Exemplo básico utilizando contato NF e bobina NA	58
FIGURA 4.9	Pisca-Pisca utilizando Set e Reset	59
FIGURA 4.10	Visualização das divisões da área de memória no software CX-Programmer	61
FIGURA 4.11	Função <i>New PLC Instruction</i>	62
FIGURA 4.12	Formas de onda do temporizador TON	63
FIGURA 4.13	Parâmetros de configuração e simbologia para temporizadores CLP Omron	64
FIGURA 4.14	Pisca-pisca utilizando temporizador de pulso	65
FIGURA 4.15	Simbologia para contadores do CLP Omron	66
FIGURA 4.16	Sistema de controle de nível de um reservatório	67
FIGURA 4.17	Diagrama de interconexões elétricas para sistema de controle de nível	68
FIGURA 4.18	Sistema de partida estrela-triângulo de um motor de indução trifásico	69

FIGURA 4.19 Fluxograma da Aplicação Partida Estrela-Triângulo de Motor Trifásico	71
FIGURA 4.20 Conexões estrela e triângulo no motor trifásico	72
FIGURA 4.21 Sistema de furação de peças semiautomático	73
FIGURA 4.22 Funcionamento dos sensores na furadeira	73
FIGURA 4.23 Fluxograma da Aplicação da Furadeira Semiautomática	75
FIGURA 4.24 Sistema de controle de um veículo em linha férrea	77
FIGURA 4.25 Diagrama de interconexões elétricas para o sistema de reversão - motor indução	79
FIGURA 4.26 Uma aplicação utilizando semáforos num cruzamento	80
FIGURA 5.1 Unidade de conservação	87
FIGURA 5.2 Diagrama de blocos de sistemas de controle pneumáticos	88
FIGURA 5.3 Cilindro de simples ação	90
FIGURA 5.4 Cilindro de dupla ação.....	90
FIGURA 5.5 Representação da válvula 3/2 vias NF, acionamento por botão, retorno por mola	92
FIGURA 5.6 Representação válvula 3/2 vias NF, acionamento por solenoide, retorno por mola	93
FIGURA 5.7 Representação da válvula 5/2 vias, acionamento por duplo solenoide	93
FIGURA 5.8 Sensores capacitivo, indutivo e ótico	95
FIGURA 5.9 Circuitos pneumáticos básicos	96
FIGURA 5.10 Uma implementação utilizando circuito pneumático	97
FIGURA 6.1 PLC Festo CPX-CEC	99
FIGURA 6.2 Software CoDeSys Versão 2.3.....	102
FIGURA 6.3 Tela inicial do Software CoDeSys Versão 2.3	102
FIGURA 6.4 Seleção do modelo do CLP	103
FIGURA 6.5 Seleção da linguagem de programação	103
FIGURA 6.6 Tela de programação em Ladder	104
FIGURA 6.7 Seleção de PLC Configuration	105
FIGURA 6.8 Seleção de “Search Network”	105
FIGURA 6.9 Números do IP dos Computadores da Rede	106
FIGURA 6.10 Selecionando o PLC ativo	107
FIGURA 6.11 Selecionando o PLC ativo – Continuação	107
FIGURA 6.12 Identificando e selecionando os módulos	108
FIGURA 6.13 Identificando e selecionando os módulos – Continuação	108
FIGURA 6.14 Verificando os endereços dos módulos de IO	109
FIGURA 6.15 Verificando os endereços dos módulos de IO – Continuação	110
FIGURA 6.16 Retornando à tela de programação	111
FIGURA 6.17 Tela de programação em Ladder	112
FIGURA 6.18 Um exemplo de programação em Ladder.....	112

FIGURA 6.19 Tela de configuração de entrada, saída ou variável	113
FIGURA 6.20 Função timer da programação em ladder no CoDeSys	114
FIGURA 6.21 Configuração do timer da programação em ladder no CoDeSys	114
FIGURA 6.22 Introdução do bloco de funções do contador na programação em ladder.....	115
FIGURA 6.23 Configuração do contador da programação em ladder no CoDeSys.....	116
FIGURA 6.24 Configuração da entrada para resetar o contador	116
FIGURA 6.25 Configuração da entrada para monitorar o valor de contagem....	116
FIGURA 6.26 Exemplo simples de utilização de contador na programação em ladder	117
FIGURA 6.27 Sistema automático de furação de peças	119
FIGURA 6.28 Sistema de identificação e contagem de peças	121
FIGURA 7.1 Configuração de hardware do CLP Modicom-Telemecanique.....	124
FIGURA 7.2 Implementação do temporizador TON no software PL7	126
FIGURA 7.3 Diagrama de tempos temporizador TON (CLP Modicom- Telemecanique)	126
FIGURA 7.4 Diagrama de tempos temporizador TOFF (CLP Modicom- Telemecanique).....	127
FIGURA 7.5 Diagrama de tempos do temporizador TP (CLP Modicom- Telemecanique)	127
FIGURA 7.6 Configuração básica do contador (CLP Modicom- Telemecanique).....	128
FIGURA 7.7 Configuração básica temporizador serie 7 (CLP Modicom- Telemecanique)	130
FIGURA 7.8 Utilização de funções de comparação no software PL7	133
FIGURA 7.9 Utilização de funções de comparação no software CX-Programmer.....	134
FIGURA 7.10 Utilização de funções de comparação e <i>flag</i> no software CX-Programmer	135
FIGURA 7.11 Exemplo de programação em linguagem ST no software CX-Programmer.....	135
FIGURA 7.12 Exemplo de código em linguagem ST no software CX-Programmer.....	136
FIGURA 7.13 Símbolos definidos para o Exemplo da Figura 7.11	136
FIGURA 7.14 Utilização de funções aritméticas no software PL7	137
FIGURA 7.15 Exemplo programação controlador PID (CLP Modicom- Telemecanique)	141
FIGURA 7.16 Tabela de animação para ajuste dos parâmetros do controlador PID	144

FIGURA 7.17 Exemplo programação função PWM (CLP Modicom- Telemecanique)	145
FIGURA 7.18 Módulo de entrada e saída MAD42	146
FIGURA 7.19 Exemplo de operação MOV para leitura em entrada e escrita em saída	147
FIGURA 7.20 Exemplo de habilit.de módulo de entrada e saída Analógica setando o LSB	148
FIGURA 7.21 Testando módulo de entrada e saída analógica	148
FIGURA 7.22 Controle PID clássico	149
FIGURA 7.23 Controle PID no PLC OMRON	150
FIGURA 7.24 Programando o controle PID no software CX-One	151
FIGURA 7.25 Bargraph com 8 lâmpadas	154
FIGURA 7.26 Sistemas para controle de velocidade	155
FIGURA 7.27 Ensaio em malha aberta no motor CC	157
FIGURA 7.28 Ensaio em malha aberta no servomotor CC	158
FIGURA 7.29 Programando o controle PID	158
FIGURA 8.1 Um exemplo de aplicação programada por grafcet	161
FIGURA 8.2 Formato e exemplo de ação	162
FIGURA 8.3 Ordem contínua sem retenção	163
FIGURA 8.4 Ordem contínua com retenção (S e R)	163
FIGURA 8.5 Ordem condicional	164
FIGURA 8.6 Ordem com retardo	164
FIGURA 8.7 Ordem limitada no tempo	165
FIGURA 8.8 Ordem impulsional	166
FIGURA 8.9 Exemplo de aplicação usando ações repetidas em etapas consecutivas	167
FIGURA 8.10 Exemplo de aplicação usando ações retentivas	168
FIGURA 8.11 Exemplo de aplicação usando etapas simultâneas	169
FIGURA 8.12 Exemplo de aplicação usando etapas separadas (OR)	170
FIGURA 8.13 Exemplo de utilização de vários tipos de receptividades na transição	171
FIGURA 8.14 Criação de primeiras etapas de um programa em SFC	172
FIGURA 8.15 Seleção de ação para uma etapa de um programa em SFC	173
FIGURA 8.16 Robô com dois graus de liberdade	174
FIGURA 8.17 Programa em Grafcet para desenvolvimento da aplicação	178
FIGURA 9.1 Aplicação com CLP em rede e ponto a ponto	180
FIGURA 9.2 Inversor de frequência e esboço de teclas de configuração	189
FIGURA 9.3 Conexões do inversor de frequência acionando motor de indução trifásico	190
FIGURA 9.4 Inversor de frequência acionando motor de indução trifásico.....	190

FIGURA 9.5	Conexão inversor de frequência em CLP Omron usando rede PROFIBUS DP	191
FIGURA 9.6	Programa desenvolvido para CLP Omron em aplicação utilizando inversor de frequência e rede PROFIBUS DP	195
FIGURA 9.7	Módulo DeviceNet DRT2-ID08.....	196
FIGURA 9.8	Conexão do módulo DeviceNet DRT2-ID08 no CLP Omron	196
FIGURA 9.9	Tela do software CX-Integrator após criação de uma rede DeviceNet.....	198
FIGURA 9.10	Aplicação Utilizando Rede DeviceNet	200
FIGURA 10.1	Ambiente de desenvolvimento (IDE) do IWS	204
FIGURA 10.2	Barra de <i>status</i> do ambiente de desenvolvimento do IWS.....	204
FIGURA 10.3	Explorador de projeto do ambiente de desenvolvimento IWS.....	205
FIGURA 10.4	Tela de sinótico de um sistema de controle de tanques de mistura.....	206
FIGURA 10.5	Criação de telas	208
FIGURA 10.6	Características das telas	208
FIGURA 10.7	Disposição das telas superior, modelo e inferior	209
FIGURA 10.8	Objetos ativos	209
FIGURA 10.9	Grupo de telas inicial com telas Superior, Modelo e Inferior com botões na tela Inferior	210
FIGURA 10.10	Objetos de dados	211
FIGURA 10.11	Tags	212
FIGURA 10.12	Inserindo classe	213
FIGURA 10.13	Inserindo vetor para tags.....	213
FIGURA 10.14	Criação de tela de Sinótico	214
FIGURA 10.15	Inserindo vetor para tags na aplicação	216
FIGURA 10.16	Criando símbolos para as variáveis.....	216
FIGURA 10.17	Configurando as propriedades do objeto.....	216
FIGURA 10.18	Agenda	217
FIGURA 10.19	Função rand	217
FIGURA 10.20	Função check box.....	218
FIGURA 10.21	Desligamento de agenda	218
FIGURA 10.22	Configurações finais de agenda	219
FIGURA 10.23	Exemplo de Tela de Sinótico em Software Indusoft Web Studio	220

LISTA DE TABELAS

TABELA 2.1	Tabela Verdade	28
TABELA 4.2	Divisão de área de memória.....	59
TABELA 7.1	Configuração de Filtros no CLP Modicom – Telemecanique.....	139
TABELA 7.2	Configuração dos Parâmetros do Controlador PID – CLP Modicom Telemecanique	143
TABELA 7.3	Configuração dos Parâmetros do Controlador PID – CLP OMRON – 1ª Parte	152
TABELA 7.4	Configuração dos Parâmetros do Controlador PID – CLP OMRON – 2ª Parte.....	153
TABELA 9.1	Comparativo entre redes Device Net, Profibus DP e AS-Interface.....	187

SUMÁRIO

1 INTRODUÇÃO AOS SISTEMAS A EVENTOS DISCRETOS	1
1.1 Classificação de Sistemas.....	2
1.1 Sistemas de Controle de Processos Contínuos.....	2
1.2 Sistemas a Eventos Discretos	4
1.3 Sistemas de Controle Híbridos.....	6
2 CONTROLADORES LÓGICOS PROGRAMÁVEIS	8
2.1 Histórico	8
2.2 Definições de Controladores Lógicos Programáveis	10
2.3 Princípio de funcionamento	11
2.4 Por que utilizar um CLP?	22
2.5 Linguagens de Programação	23
2.6 Ferramentas para o Desenvolvimento de Aplicações.....	27
3 PROGRAMAÇÃO EM LADDER	32
4 PROGRAMAÇÃO EM LADDER NO CLP OMRON	49
4.1 Controlador Lógico Programável CJ2M – Omron	49
4.2 Instruções para programação utilizando software CX-One 35	51
4.3 Programação de temporizadores e contadores.....	59
4.4 Exercícios de Fixação	66
5 PNEUMÁTICA	82
5.1 Introdução à Pneumática	82
5.2 Propriedades do Ar Comprimido	82
5.3 Sistemas de Produção e Distribuição de Ar Comprimido	84
5.3.1 Compressores	85
5.3.2 Unidade de Preparação, Resfriamento e Distribuição de Ar Comprimido	86
5.4 Estruturas de Sistemas de Controle Pneumáticos	88
5.5 Atuadores	89
5.6 Válvulas Direcionais.....	91
5.7 Sensores.....	94
5.8 Exemplos de Circuitos Pneumáticos e Eletropneumáticos	95
6 PROGRAMAÇÃO DO CLP CPX-CEC FESTO®.....	99
6.1 Hardware do Controlador Lógico Programável CPX-CEC	99
6.2 Programação do Controlador Lógico Programável CPX-CEC.....	100
6.3 Programação de Temporizadores e Contadores e Outras Funções no CLP Festo Usando CodeSys.....	113
6.4 Exercícios de Fixação	117

7 FUNÇÕES ESPECIAIS EM CLPs MODICOM-TELEMECANIQUE (SCHNEIDER ELECTRIC, FESTO e OMRON).....	123
7.1 Programação do PLC Modicom-Telemecanique no Software PL7.	123
7.2 Registradores.....	130
7.3 Comparadores.....	132
7.4 Operações Aritméticas	137
7.5 Algumas funções especiais no PLC Modicom-Telemecanique	138
7.6 Funções especiais no PLC OMRON	146
7.7 Exercícios de Aplicações.....	154
8 PROGRAMAÇÃO EM SFC / GRAFCET	160
8.1 Grafcet e SFC.....	160
8.2 Ações	162
8.3 Receptividades	170
8.4 Programação em SFC no CLP Omron	171
8.5 Exemplo de Aplicação	174
9 REDES INDUSTRIAIS	179
9.1 Redes Industriais para Automação Industrial.....	179
9.2 Redes Profibus DP	188
9.3 Rede DeviceNet	195
9.4 Exemplo: Uma Aplicação Utilizando Rede DeviceNet	198
10 SISTEMAS DE SUPERVISÃO E AQUISIÇÃO DE DADOS	201
10.1 Sistema SCADA.....	201
10.2 Indusoft Web Studio.....	202
10.3 Exemplo: Aplicação de Controle em Tanques de Mistura	205
REFERÊNCIAS BIBLIOGRÁFICAS	221
ANEXO.....	224

1 – INTRODUÇÃO A SISTEMAS A EVENTOS DISCRETOS.

O livro "Automação Industrial Utilizando PLC`s" tem como objetivo apresentar a utilização e programação de controladores lógicos programáveis (CLP) em aplicações de automação flexível. O enfoque está centrado principalmente na automação de sistemas a eventos discretos, mas mostra também os fundamentos de aplicações de automação de processos contínuos utilizando CLP's.

Em termos de aplicações de automação, dependendo do grau de flexibilidade, podemos definir dois tipos de sistemas de automação (Figura 1.1):

- automação rígida: utiliza lógica fixa, sendo necessário modificar componentes elétricos ou pneumáticos para realizar modificações;
- automação flexível: utiliza lógica programável, por exemplo com CLP's, de forma que alterações consideráveis possam ser realizadas por software.

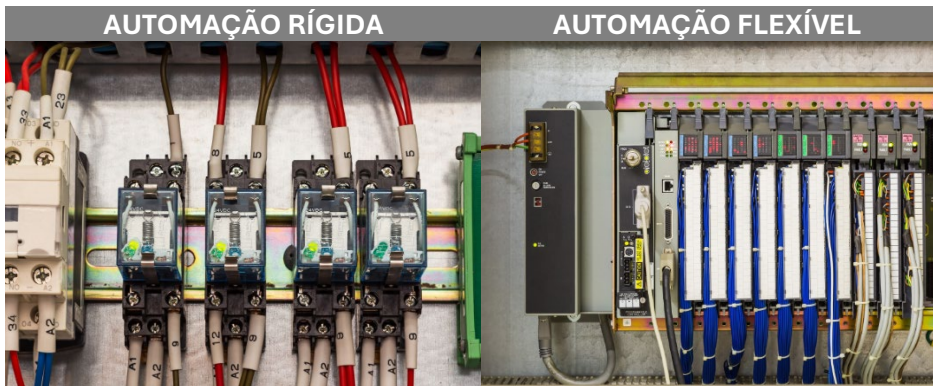


Figura 1.1 – Automação rígida e automação flexível

1.1 – Classificação de Sistemas.

Dependendo dos tipos de entradas e saídas controladas os sistemas de controle convencionais podem ser classificados nas seguintes categorias:

- sistemas de controle de processos contínuos;
- sistemas a eventos discretos;
- sistemas híbridos.

1.1.1 – Sistemas de Controle de Processos Contínuos.

Os sistemas de controle de processos contínuos, ou simplesmente, sistemas a variáveis contínuas (SVC) são sistemas cujos sinais de entrada e saída são contínuos no tempo.

A Figura 1.2 ilustra instrumentos para medição de variáveis contínuas.

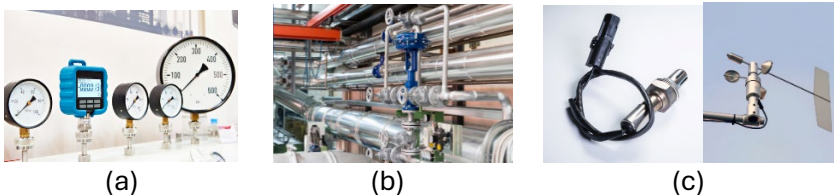


Figura 1.2 – Instrumentos para medição de variáveis contínuas

(a) Manômetros (b) Válvula de controle (c) Medidores de nível, vazão e temperatura

Os sistemas a variáveis contínuas geralmente envolvem a utilização de dispositivos reguladores desenvolvidos de várias formas tais como controladores industriais, controladores PID, sistemas de aquisição de dados, SDCD's, etc. (Figura 1.3).

Cap.1 –Introdução a Sistemas a Eventos Discretos



(a)



(b)

Figura 1.3 – Exemplos de sistemas utilizados no controle de SVCs
(a) Controladores industriais (b) Sistema de aquisição de dados

Existe uma grande infinidade de aplicações onde o controle de processos industriais é utilizado. Podemos citar, por exemplo, aplicações em sistemas de geração de energia elétrica (hidrelétricas, caldeiras, usinas eólicas), nas indústrias de óleo e gás, petroquímica, e de papel e celulose, na automação de máquinas elétricas (controle de posicionamento ou velocidade), etc.

As aplicações mais tipicamente encontradas são os sistemas de controle de nível, vazão e temperatura de fluidos, geralmente comandados por uma central de controle com SDCCD (Sistema Digital de Controle Distribuído) como ilustrado na Figura 1.4.



Figura 1.4 – Central de controle de um SDCCD

1.1.2 – Sistemas a Eventos Discretos.

Os sistemas a eventos discretos (SED) ou sistemas de manufatura são sistemas cujos sinais de entrada e saída são discretos no tempo. Isto significa que as entradas e saídas possuem estados definidos. De um modo geral, nestes sistemas, o estado da saída somente irá alterar para outro estado se uma lei de controle determinar a mudança do mesmo. A variável controlada pode ser uma grandeza contínua qualquer – pressão, vazão, nível, tensão, corrente, etc. – desde que o comportamento seja caracterizado por estados bem definidos no tempo. Por isto este controle é tipicamente conhecido como sistema ON-OFF. Os sistemas de controle a eventos discretos (SED) envolvem a utilização de dispositivos de controle discretos como CLP's, lógica a relés, sistemas SCADA (usualmente constituído de sistema supervisorio que permite desenvolver a Interface Homem Máquina (HMI), unidades remotas conectadas em sensores, infraestrutura de comunicação (protocolos de comunicação) e PLC's.

As aplicações ocorrem nas mais diversas áreas (Figura 1.5), sendo a maioria delas controlados por Controladores Lógicos Programáveis (CLPs). Podemos citar algumas.

- Controle de robôs na indústria automotiva – neste caso os robôs são utilizados na soldagem, pintura, fixação de peças, furação, etc.
- Controle de robôs ou braços mecânicos com o objetivo de executar operações repetitivas, efetuar a movimentação de materiais, efetuar a soldagem de componentes eletrônicos em placas de circuito impresso, dentre outros.
- Desenvolver sistemas de manufatura, sistemas automatizados compostos por múltiplos subsistemas, tais como fabricação, montagem, transporte e armazenagem. Exemplos de aplicações podem ser vistos na indústria de papel e celulose, de bebidas e alimentos, farmacêutica, mecânica, etc.
- Automação de Sistemas de Embalagem.
- Controle de sistemas pneumáticos e hidráulicos;

Cap.1 –Introdução a Sistemas a Eventos Discretos

- Alarmes, controle de acesso e sistemas de segurança instrumentados (SIS); SIS são constituídos de sensores e elementos atuadores comandados geralmente por PLCs com o objetivo de levar o processo a um estado seguro quando condições pré-determinadas forem violadas; sistema de intertravamento.
- Elevadores, balanças eletrônicas
- Controle de tráfego (semáforos, metrô e trens).



Figura 1.5 – Aplicações de sistemas a eventos discretos tipicamente controlados por CLPs ou outros sistemas microprocessados

Em outras aplicações os sistemas a eventos discretos são controlados por microcontroladores ou circuitos microprocessados (Figura 1.6). Podemos citar, por exemplo, as seguintes aplicações:

- automação industrial e residencial incluindo controle de iluminação, som, vídeo, temperatura, sistema de segurança e controle de acesso baseado em controle por voz, reconhecimento da íris ou digital, acionamento de teclas do celular, etc.
- sistemas de usinagem de peças em máquinas CNC onde a engenharia, projeto e manufatura podem ser desenvolvidos por computador (CAE/CAD/CAM);
- aplicações na medicina, como por exemplo o desenvolvimento de próteses robóticas e membros artificiais, dispositivos auxiliares para movimentação e fisioterapia, órgãos artificiais, etc.;

Cap.1 –Introdução a Sistemas a Eventos Discretos

- aplicações de nanotecnologia;
- robôs para entretenimento e brinquedos eletrônicos.



Figura 1.6 – Outras aplicações de sistemas a eventos discretos

A utilização de sistemas a eventos discretos envolve além disso a aplicação de diversas tecnologias de automação como por exemplo conceitos de CIM – Manufatura Integrada por Computador, MES – *Manufacturing Execution System*, ERP – *Enterprise Resources Planning* (Gestão), softwares supervisórios (SCADA – *Supervisory Control and Data Acquisition*), IHM – Interface Homem-Máquina, protocolos de comunicação (MODBUS, PROFIBUS), etc.

1.1.3 – Sistemas de Controle Híbridos.

Os sistemas de controle híbridos ou semi-contínuos englobam as características de sistemas contínuos e discretos. Podemos citar como exemplo, o fermentador na indústria química cujo processo de operação envolve etapas discretas (enchimento, drenagem e limpeza) e etapas contínuas (aquecimento e fermentação).

As principais aplicações referem-se a sistemas de controle de batelada em indústria de alimentos e bebidas, química, farmacêutica, etc., como mostrado na Figura 1.7.

São também exemplos de aplicação de sistemas híbridos os sistemas que envolvem processamento de imagem onde geralmente são monitoradas variáveis contínuas e discretas de forma a controlar os

Cap.1 –Introdução a Sistemas a Eventos Discretos

dispositivos. São exemplos de aplicação a seleção automática de frutos e vegetais por forma e cor, detecção de defeitos e classificação de produtos, verificação do nível de envasamento de frascos e garrafas, verificação da presença e correção de informações impressas em caixas de alimentos, verificação do fechamento incorreto ou incompleto de embalagens, ausência de produtos ou quantidade incorreta de produtos em embalagens, verificação do nível de envasamento de frascos e garrafas, verificação da inserção e fechamento corretos de tampas, identificação de elementos estranhos no interior de frascos e garrafas, etc.



Figura 1.7 – Sistema de controle em batelada na indústria química

2 – CONTROLADORES LÓGICOS PROGRAMÁVEIS.

Historicamente, os estudos relativos à modelagem e controle de sistemas a estado contínuo são anteriores aos de sistemas a estado discreto. Sob a perspectiva comportamental, os modelos podem ser subdivididos em transformativos e reativos. Os modelos transformativos ou a estado contínuo descrevem o sistema através de equações elementares e de suas interconexões, sendo estas equações definidas para todo o tempo (modelos contínuos no tempo) ou definidas ou usadas em pontos discretos no tempo (modelos discretos no tempo). Exemplos destes modelos são a Função de Transferência e a descrição por Variáveis de Estado, enquadrados na Teoria de Controle Moderno e Controle Digital.

Os modelos reativos ou a estado discreto representam os possíveis estados que o sistema pode assumir, sendo a mudança de estado e as saídas produzidas pelo sistema decorrentes da combinação lógica de entradas (eventos e condições) e do estado em que o sistema se encontra. Exemplos destes modelos são a Rede de Petri, o diagrama de transição de estados (diagrama de estados) e o diagrama Grafcet. Estes modelos mostram explicitamente a sequência com que ocorrem os estados e saídas, sem estabelecer normalmente uma relação direta com o tempo.

Neste e nos dois próximos capítulos trataremos explicitamente dos controladores lógicos programáveis aplicados a sistemas a estados discretos. Serão considerados os aspectos de modelagem, hardware de controladores lógicos programáveis, linguagens típicas de programação e aplicações (NATALE, 1997) (SILVEIRA e SANTOS, 1998).

2.1 – Histórico

Alguns pesquisadores consideram o ano de 1804, quando Jacquard inventou a máquina de tear com cartões perfurados, o 'início' do controle de sistemas sequenciais que são uma classe de sistemas a eventos discretos. Entretanto, antes disso, James Watt, em 1788,

desenvolveu os primeiros reguladores mecânicos. Seguiram-se a instrumentação e reguladores do tipo pneumático e hidráulico e, após as guerras mundiais, do tipo eletrônico com tecnologia analógica.

Ainda no século XVIII já existem registros de uma máquina de tear automática com cartões perfurados (~1790-1801) e de uma moenda automática por esteiras (~1791). Em ~1824, Sturgeon desenvolveu o eletroímã que permitiu a Henry construir o relé eletromagnético em ~1836.

A Álgebra de Boole, que é uma das bases matemáticas do controle de sistemas a eventos discretos, foi proposta por Boole em 1854 e, em 1936, Stibitz desenvolve a primeira calculadora eletrônica feita à base de relés.

A modelagem e controle de sistemas a estado discreto é relativamente recente. Citam-se ao leitor, como referências indicativas, algumas datas. A Rede de Petri, base para o diagrama Grafcet (expressivo para modelagem de sistemas a eventos discretos), deriva dos trabalhos de C. A. Petri no início dos anos 60. A teoria de sistemas a eventos discretos, baseia-se nos estudos de autômatos finitos e linguagens formais desenvolvidos ao longo deste século, sobretudo nesta segunda metade. O teorema de Kleene, por exemplo, que estabelece uma importante correlação entre a teoria de autômatos finitos e linguagens formais no estudo de sistemas a eventos discretos, data de 1950 aproximadamente.

O controle baseado na tecnologia de computadores, o Controle Programável (MIYAGI, 1996), vem sendo cada vez mais utilizado na indústria. Sua aplicação destina-se ao controle de Sistemas a Eventos Discretos (SED) em substituição a sistemas de controle baseados em relés eletromagnéticos. A estratégia de controle consiste na execução de um programa que define a evolução dos processos. Como o programa fica armazenado numa memória interna do controlador, qualquer alteração dos procedimentos pode ser realizada facilmente via software. Atualmente encontram-se disponíveis diversos tipos de controladores lógicos programáveis de vários fabricantes capazes de implementar com eficiência o controle de sistemas a eventos discretos.

Os controladores lógicos programáveis (CLP) surgiram na década de 60 na indústria automobilística para substituir painéis de controle a relés. Na década de 80, com a introdução dos microcontroladores, passaram a contar com maior capacidade de executar algoritmos de controle complexo e de conectividade em rede (BATTEN JR, 1994). Atualmente os controladores lógicos programáveis incorporam todos os tipos de funções necessárias para a realização do controle de SED. Além disso as funções de comunicação do CLP foram aperfeiçoadas permitindo sua aplicação dentro de um sistema de controle em rede, onde são integradas as técnicas de controle de sistemas a eventos discretos, controle de sistemas contínuos (manipulando, portanto, variáveis analógicas) e processamento de informações para gerenciamento industrial através de sistemas supervisórios. Estes controladores com muito mais recursos são comumente referenciados por alguns autores como Controladores Programáveis (CP).

2.2 – Definições de Controladores Lógicos Programáveis

Os primeiros CLPs foram desenvolvidos para controlar dispositivos na indústria automobilística. Por isto a primeira definição para CLP dizia que “Controlador Programável é um microcomputador destinado ao controle de processos industriais sendo que sua função básica é a de substituir uma lógica feita por relés por uma lógica armazenada em uma memória de programa”. De fato, nas primeiras aplicações o CLP era utilizado com o objetivo de substituir os enormes painéis a relés por um terminal programável onde a lógica pudesse ser desenvolvida. Exatamente por isto, a primeira linguagem de programação utilizada foi a linguagem Ladder, que tem o aspecto similar à representação dos circuitos eletrônicos de controle tipicamente utilizados na representação de sistemas de potência e proteção incluindo relés, contadores e bobinas.

A definição da ABNT é mais genérica e diz que: “Controlador Programável é um equipamento eletrônico digital com hardware e software compatíveis com aplicações industriais”.

2.3 – Princípio de Funcionamento

Um CLP é basicamente composto por dois elementos principais: uma CPU e circuitos de interface para os sinais de entrada e saída (Figura 2.1).

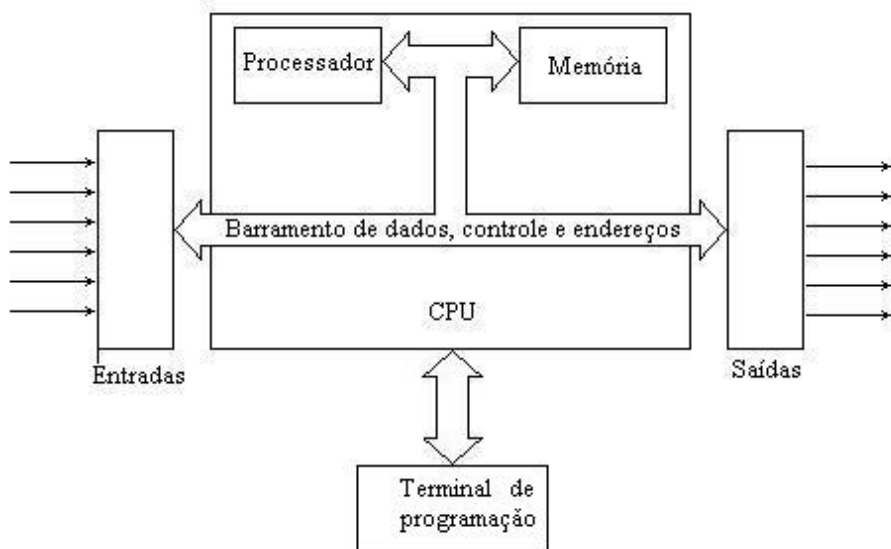


Figura 2.1 – Diagrama de blocos de um CLP

O princípio fundamental de funcionamento de um CLP é a execução por parte da CPU de um programa conhecido como “executivo” e de responsabilidade do fabricante, que realiza ciclicamente as ações de leitura das entradas, execução do programa de controle do usuário e atualização das saídas (Figura 2.2).

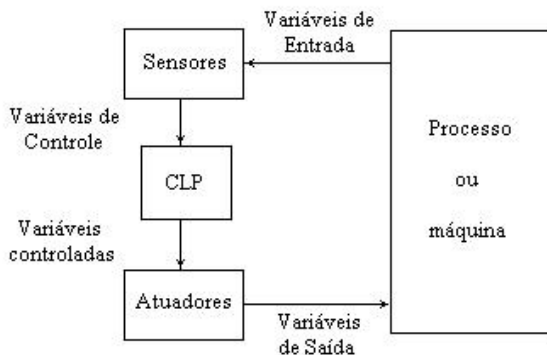


Figura 2.2 – Ciclo de varredura

O tempo para execução dessas tarefas, chamado ciclo de varredura ou *scanning*, depende, dentre outros fatores, da velocidade e característica do processador utilizado, do tamanho do programa de controle do usuário, além da quantidade e tipo de pontos de entrada / saída. Como regra geral, tal tempo se encontra na faixa média de milissegundos (até microssegundos nos CLPs de última geração).

A especificação de um projeto com CLPs não é uma tarefa trivial. De fato, dentre os vários modelos disponíveis no mercado (Figura 2.3), nenhum fabricante possui uma opção que seja a melhor solução para todas as aplicações.

Em muitas vezes, a opção por um ou outro fabricante depende do país de origem dos equipamentos ou outras questões históricas. Por exemplo, na área automotiva:

- Volkswagen (empresa de origem alemã) – utilizam principalmente CLP's da Siemens, e em alguns casos ABB;

Cap. 2 – Controladores Lógicos Programáveis

- GM, Ford (sede em Detroit) – utilizam principalmente equipamentos da Rockwell / Allen Bradley, mas também Siemens;
- Scania (empresa de origem sueca) – utilizam tipicamente Siemens, ABB, Rockwell / Allen Bradley;
- Peugeot / Citroen – utilizam principalmente equipamentos da Schneider;
- Honda, Nissan – utilizam produtos da Mitsubishi, Toshiba, Yokogawa e Omron (Figura 2.4);
- Fiat (Itália) – utilizam principalmente Rockwell e Siemens.



Figura 2.3 – Alguns modelos de controladores lógicos programáveis

Alguns fabricantes são líderes de mercado na área de acionamentos, como por exemplo ABB e WEG.

Outros fabricantes que merecem destaque na área: Atos, GE Fanuc, Altus, etc.

Quanto à conexão em rede, cada fabricante estabelece um protocolo para fazer com que seus equipamentos troquem informação entre si. Os protocolos mais comuns são Modbus (Modicon – Schneider

Electric), EtherCAT (Beckhoff), Profibus (Siemens), Unitelway (Telemecanique – Schneider Electric) e DeviceNet (Allen Bradley), entre muitos outros.



Figura 2.4 – Alguns modelos de controladores lógicos programáveis OMRON

Ao selecionar um modelo deve-se atentar para detalhes como:

- qual o equipamento atende plenamente às minhas necessidades, com a melhor relação custo-benefício;
- quais as especificações necessárias para o meu equipamento.

É necessário, além disso, conhecer profundamente a aplicação a que o equipamento está destinado. Para isto deve-se fazer um levantamento completo dos sensores, atuadores e instrumentos de forma a definir:

- a precisão na medição (tipos de sinais e níveis de tensão);
- as distâncias envolvidas (limitantes para a especificação do tipo de rede de automação)
- os locais de instalação de IHMs, SDCDs, painéis, etc.;
- as tensões e potências disponíveis; etc.

Os aspectos que devem ser considerados na seleção de um CLP para automação são:

- **número de entradas e saídas analógicas e digitais (I/O);**
- **expansibilidade;**
- **recursos especiais** disponíveis;
- **desempenho** tal como tempo de *scan* ou tempo aceitável para varredura no sinal de entrada, presença de entradas e saídas imediatas (não dependentes do *scan*); velocidade de comunicação entre CLPs;
- capacidade de interconexão com **redes de automação** como por exemplo AS-i, CAN, ControlNet, DeviceNet, Ethernet/IP, Fieldbus, FIP, Hart, Interbus, Modbus, Profibus DP/PA/FMS entre outras;
- **confiabilidade;** adoção de normas de garantia de qualidade e certificação para uso em ambiente industrial;
- **tamanho e peso;**
- **custo do equipamento, acessórios e softwares;** além do custo de aquisição do equipamento deve-se considerar o custo de ferramentas especiais para montagem de cabos, custo de cabos e conectores, custo de borneiras, custo de equipamentos de configuração de redes de automação; custo de softwares de programação de CLPs, softwares de configuração de redes de automação e softwares de sistemas de supervisão e controle;
- **linguagens de programação** disponíveis e conversão de códigos;
- **vida útil,** pois alguns fabricantes não prestam assistência para modelos antigos ao lançarem novos produtos sendo, portanto, um grande risco ao investir em equipamentos obsoletos;
- **custo de mão-de-obra** para desenvolvimento dos softwares de CLPs, SDCDs e sistemas de supervisão e controle;
- **suporte técnico** para *start-up* e treinamento e assistência técnica;
- **custo de assistência técnica;** disponibilidade de atendimento; tempo de manutenção
- Em relação à **expansibilidade** deve-se atentar que a introdução de I/O adicionais, espaço em painéis, integração com redes de automação (AS-i, CAN, DeviceNet, Fieldbus, Hart, Interbus, Modbus, Profibus ...) permite expandir a capacidade de I/Os dos

CLPs, possibilitando que equipamentos de pequeno porte possam controlar uma quantidade de I/Os muito grande. A possibilidade de expansão de memória também deve ser considerada.

A Figura 2.5 ilustra a expansão de I/O's em um PLC modular da empresa OMRON.

Em relação aos recursos especiais disponíveis podemos citar, por exemplo, contadores rápidos, função PWM para aplicações de posicionamento, leitura de temperatura, relógio de tempo real, portas de comunicação adicionais, IHM incorporada, PID, sintonia automática, cálculos em ponto flutuante, cálculos avançados (SEN, CON, TAN, LOG, EXP, BCD, HEX, ASCII,...), etc.

Na indústria o CLP utiliza uma arquitetura modular composta de fonte, CPU e módulos de entrada e saída, além de módulos para comunicação em rede. Os módulos de entrada / saída podem ser digitais com 8, 16 ou 32 pontos ou analógicos de 2, 4, 8 ou 16 pontos. Para manipular dados analógicos alguns CLPs incorporam conversores A/D ou D/A.

Alguns exemplos de dispositivos de entrada e saída comumente utilizados na indústria são listados abaixo.

- Entradas Digitais (Figura 2.6) – Chaves seletoras, *push-buttons*, sensores fotoelétricos, chaves fim de curso (detector por contato), sensores de proximidade, chaves sensores de nível, contatos de partida e contatos de relés.
- Saídas Digitais (Figura 2.7) – Relés de controle, solenoides, partida de motores, válvulas, ventiladores, alarmes, lâmpadas, sirenes. Algumas saídas multibits: motor de passo, display de sete segmentos e alfanuméricos.

Cap. 2 – Controladores Lógicos Programáveis

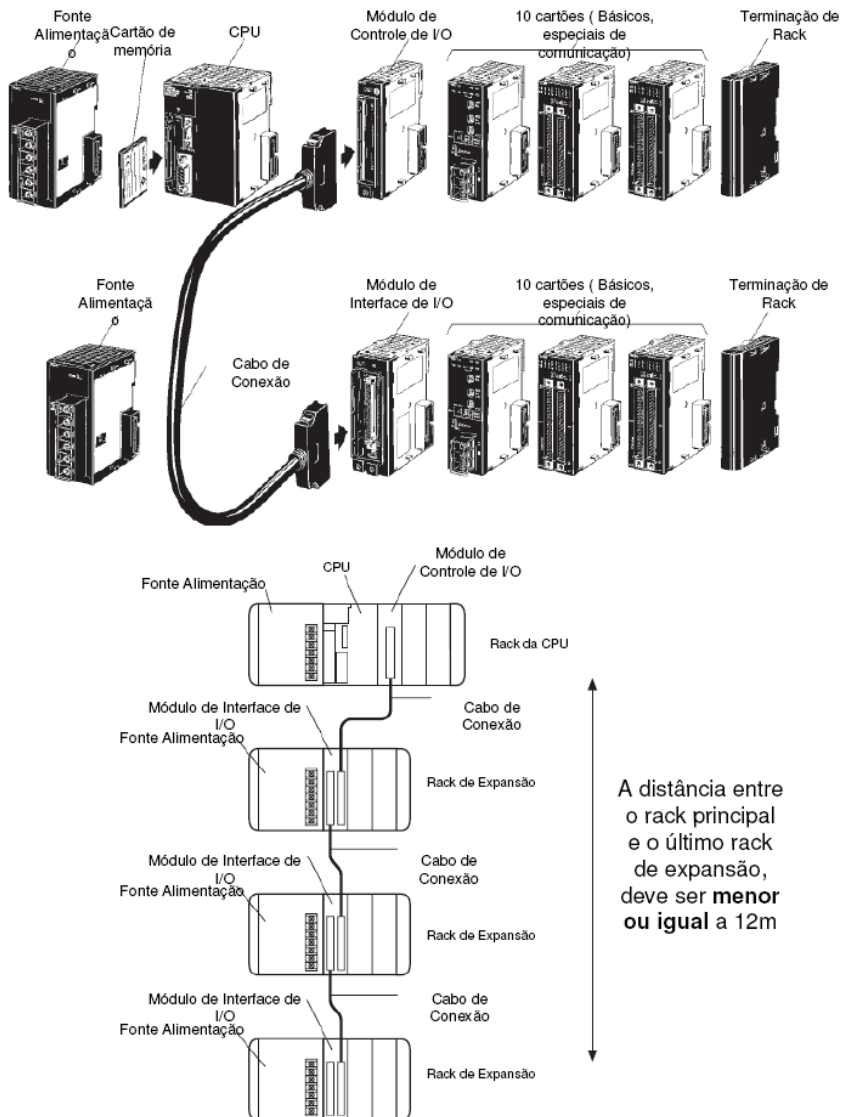


Figura 2.5 – Expansão de I/O's utilizando sistema de expansão de racks

Cap. 2 – Controladores Lógicos Programáveis

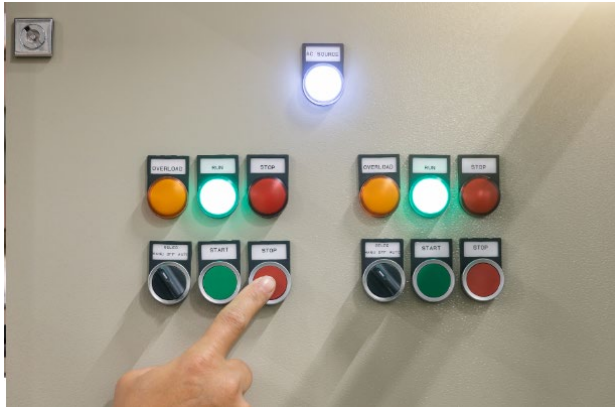


Figura 2.6 – Entradas digitais

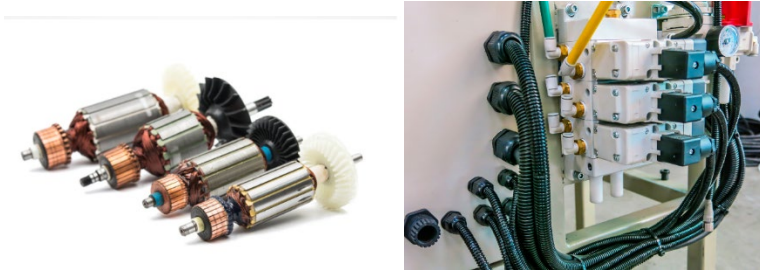


Figura 2.7 – Saídas digitais

- Entradas Analógicas – Transdutores de tensão e corrente, transdutores de temperatura, transdutores de pressão, transdutores de fluxo, potenciômetros, L.V.D.T. (*Linear Variable Differential Transformer*) (Figura 2.8).
- Saídas Analógicas – Válvula analógica (Figura 2.8), acionamento de motores, controladores de potência, atuadores analógicos, mostradores gráficos, medidores analógicos.

Cap. 2 – Controladores Lógicos Programáveis

- Outros tipos de I/Os – Encoders (geralmente apresentam entradas digitais multibits), PWM, termopares, PT100).

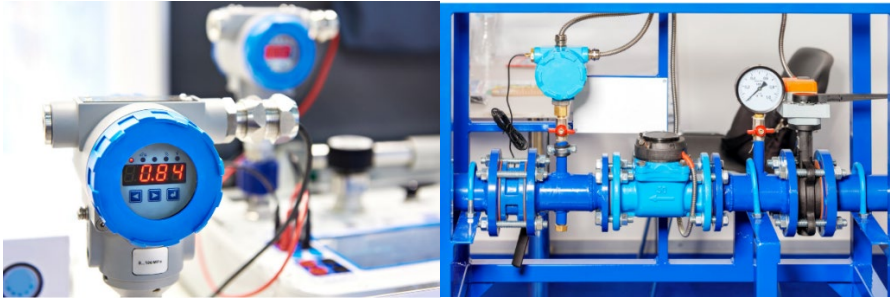


Figura 2.8 – Entradas e saídas analógicas

Algumas características relacionadas às entradas e saídas devem também ser consideradas tais como níveis de tensão, capacidade da corrente de saída, isolamento, etc. Neste sentido podemos verificar os seguintes aspectos:

- número de entradas digitais para cada tensão DC (5, 12, 24, 48 Vdc) (Figura 2.9); AC (110 / 220 Vac) (Figura 2.10);

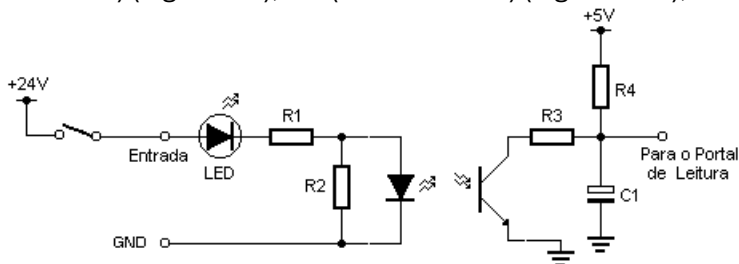


Figura 2.9 – Entrada DC - Circuito de acoplamento óptico para entradas digitais - CLP OMRON

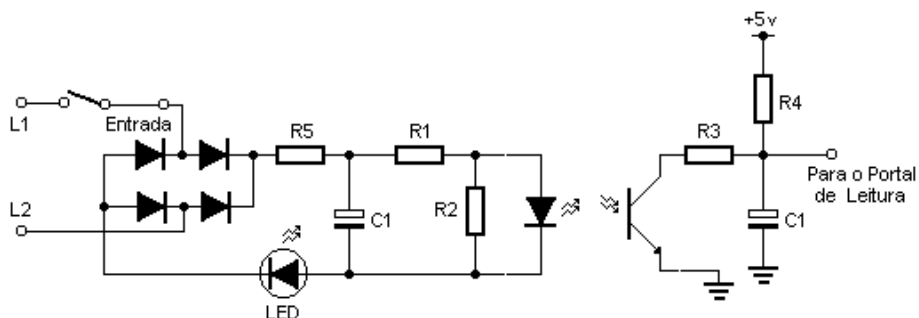


Figura 2.10 – Entrada AC - Circuito de acoplamento óptico para entradas digitais - CLP OMRON

- número de saídas digitais para cada tensão - DC (5, 12, 24, 48 Vdc); AC (110 / 220 Vac); saídas digitais à transistor PNP ou NPN (Figura 2.11), saídas à triac (Figura 2.12) ou saídas digitais à relé (Figura 2.13); as saídas a relé somente devem ser usadas para aplicações em baixa frequência (máximo 10 Hz); vida útil da ordem de 1.000.000 de operações;

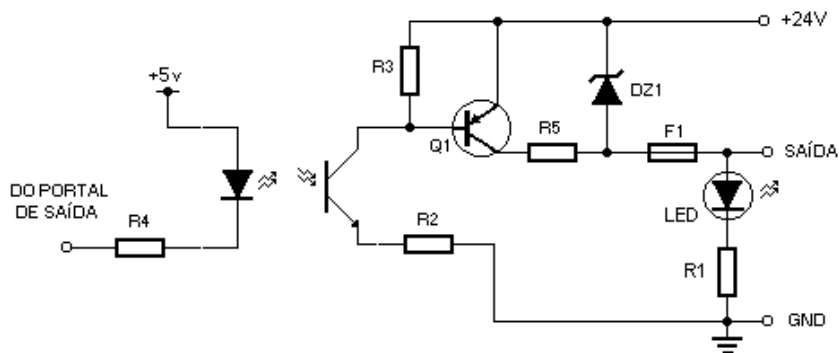


Figura 2.11 – Saída digital a transistor PNP - Circuito de acoplamento óptico - CLP OMRON

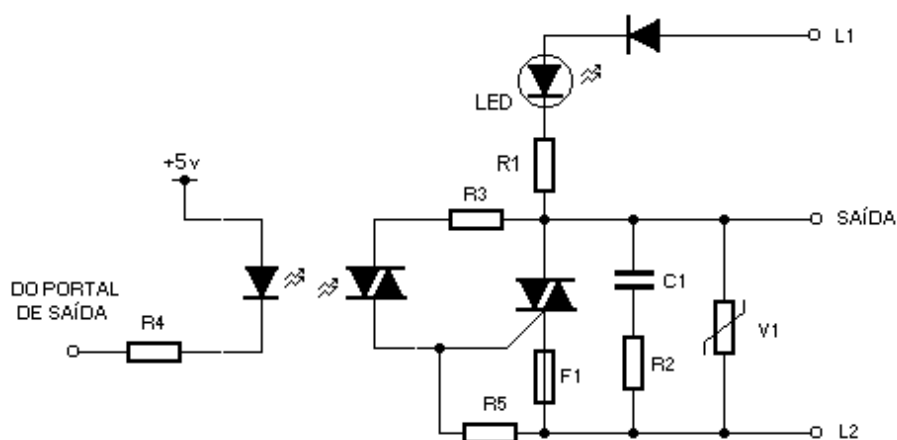


Figura 2.12 – Saída digital AC a Triac - Circuito de acoplamento óptico - CLP OMRON

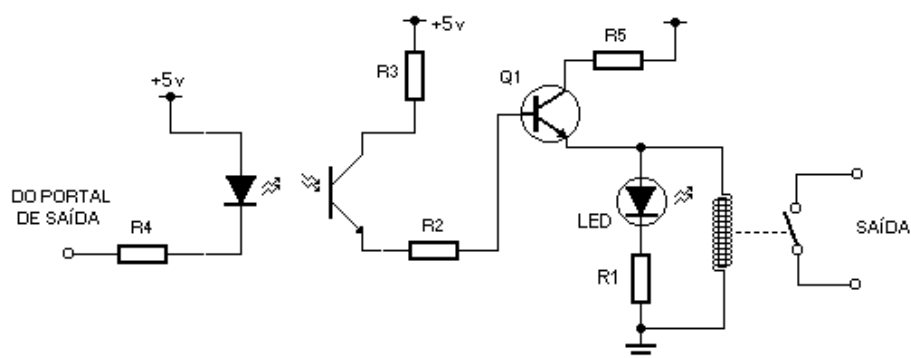


Figura 2.13 – Saída digital a relé - Circuito de acoplamento óptico - CLP OMRON

- número de entradas e saídas com isolamento (isolação elétrica dos sinais digitais e isolamento galvânica dos sinais analógicos);

- número de entradas analógicas em corrente (0..20, 4..20 mA);
- número de entradas analógicas em tensão (0..5, 0..10, -10..10 Vdc);
- número de entradas analógicas em corrente (0..20, 4..20 mA);
- número de entradas analógicas em tensão (0..5, 0..10, -10..10 Vdc);
- resolução dos I/Os analógicos.

Outro aspecto importante que deve ser considerado é a memória necessária em função do número de I/Os disponíveis. Em projetos simples deve-se ter um mínimo de 20 *words* (programa + dados) por I/O. Consideramos projetos simples aqueles que envolvem lógicas de controle simples, normalmente sem utilizar sistema de supervisão ou utilizando IHM com poucas informações e que inclui geralmente poucos pontos de I/Os analógicos. Em projetos complexos deve-se ter no mínimo 50 *words* (programa + dados) por I/O. Projetos complexos envolvem lógicas de controle complexas, com malhas de controle PID, intertravamentos complexos, cálculos avançados, conexão de IHM com muitos recursos ou à um sistema de supervisão e controle, grande número de I/Os analógicos.

2.4 – Por Que Utilizar um CLP?

Os CLPs são a melhor solução na automação de processos que apresentam relativa complexidade e envolvem um grande número de entradas e saídas, principalmente quando o número de entradas e saídas digitais são predominantes. Quando o sistema é predominantemente constituído somente de plantas de controle analógicas uma solução alternativa é utilizar controladores programáveis, tais como os controladores PID ou sistemas de aquisição de dados.

Os CLPs são utilizados na automação de sistemas principalmente com os seguintes objetivos:

- aumento de produtividade;

- automatização e redução de custos;
- otimização da utilização de recursos e economia de matéria-prima;
- segurança de pessoas e instalações;
- garantia de segurança em aplicações críticas.

Em alguns casos, principalmente quando o processo envolve riscos para os seres humanos ou para o meio ambiente, a segurança é um quesito primordial. Nestas situações deve-se assegurar que os riscos sejam mantidos em níveis aceitáveis (sem comprometimento da vida, qualidade de vida ou recursos ambientais) mesmo na ocorrência de falhas de equipamento, de operação ou acidentes. Os níveis de integridade das instalações (SIL – *Safety Integrity Level*) são regulamentados pela norma ANSI/ISA S84. Além disso, a norma IEC 615108 mostra métodos para cálculo do desempenho de segurança e confiabilidade de um sistema.

2.5 – Linguagens de Programação

A programação dos CLPs é efetuada utilizando softwares instalados em computadores, sendo a comunicação através de uma porta serial ou placa colocada no *slot* do computador.

Para os CLPs que dispõem de recurso de programação *off-line* não é necessário que o programador esteja conectado a eles para desenvolver o software, porém a conexão é fundamental para ler/enviar o programa, durante a colocação em funcionamento do software e para as atividades de manutenção.

Com o propósito de estabelecer um padrão no modo pelo qual os softwares de programação pudessem processar seus comandos para controlar dispositivos de controladores lógicos programáveis um grupo de trabalho foi formado dentro da *International Electrotechnical Commission* (IEC), organização normativa internacional com base em Genebra, formada por representantes de todo o mundo. O grupo de trabalho foi formado em 1979 para avaliar o projeto completo de controladores

programáveis, incluindo hardware, instalação, teste, documentação, programação e comunicação (FONSECA *et al.*, 2008). Esta norma é constituída das seguintes partes:

Parte 1 – *General Information* – Definição da terminologia e dos conceitos (2ª edição publicada em 2003);

Parte 2 – *Equipment Requirements and Tests* – Teste de verificação e fabricação eletrônica e mecânica (3ª edição publicada em 2007);

Parte 3 – *Programming Languages* – Estrutura do software do controlador programável, linguagens e execução de programas (2ª edição publicada em 2003);

Parte 4 – *User Guidelines* – Orientações para seleção, instalação e manutenção de controladores programáveis (2ª edição publicada em 2004);

Parte 5 – *Communications* – Funcionalidades para comunicação com outros dispositivos (1ª edição publicada em 2000);

Parte 6 – *Reservada* – Inicialmente prevista para comunicação via Fieldbus, contudo com a aprovação da Norma IEC 61158, definindo oito padrões de rede Fieldbus, a IEC decidiu cancelar o desenvolvimento da parte 6, deixando-a reservada para uso futuro;

Parte 7 – *Fuzzy Control Programming* – Funcionalidades de software, incluindo blocos funcionais padrões para tratamento de lógica nebulosa dentro de controladores programáveis (1ª edição publicada em 2000);

Parte 8 – *Guidelines for the Application and Implementation of Programming Languages* – Orientações para implementação das linguagens IEC 61131-3 (2ª edição publicada em 2003).

A norma IEC 61131 era denominada, inicialmente, IEC 1131. Recentemente a IEC alterou a forma de codificação de suas normas, incluindo mais um algarismo. Desta forma, a norma 1131 passou a ser denominada IEC 61131. Portanto, ambas as denominações se referem à mesma norma.

A norma IEC 61131-3 reconhece dois grandes blocos que representam o tipo de linguagem de programação utilizada, cada qual contendo um conjunto de linguagens correntes (FONSECA *et al.*, 2008). Apesar da norma IEC 61131-3 padronizar diversos aspectos relativos à programação de controladores programáveis, muitos desses são dependentes de implementação e outros não são definidos. Desta forma, não se pode esperar que os produtos de mercado sejam totalmente compatíveis entre si, a ponto de permitir que uma das principais vantagens da adoção da norma seja alcançada: a portabilidade, ou seja, a capacidade de intercâmbio de programas entre sistemas de fabricantes diferentes. A norma IEC apresenta uma tabela de características padrões e exige apenas que os fabricantes indiquem como os produtos são aderentes a estas características para serem considerados controladores compatíveis com a IEC. A intenção proposta pela IEC para garantir que a norma seja seguida não é o bastante para que os fabricantes possam aderir plenamente ao padrão, até mesmo porque existem erros de interpretação e implementação, sendo necessários testes para certificação de produtos que são executados por comitês específicos. Atualmente, apenas algumas especificações e testes de certificação estão disponíveis, fazendo com que o número de produtos certificados seja modesto. Mas já é perceptível a tendência dos fabricantes em atender às características da norma IEC 61131, principalmente no tocante às linguagens de programação, recursos multitarefa e reutilização de software. Portanto os usuários já podem usufruir dos principais benefícios proporcionados pela norma e começam a considerar a adoção da norma como um ponto determinante para os novos sistemas de controle.

A norma IEC 61131-3 define cinco linguagens de programação sendo três gráficas e duas textuais. São elas:

Linguagens Gráficas

- Diagrama de Funções Sequenciais (SFC – *Sequential Function Chart*) $\Rightarrow$ Grafcet;
- Diagrama em Ladder (LD – *Ladder Diagram*) $\Rightarrow$ Diagrama de Contatos;
- Diagrama de Blocos de Funções (FBD – *Function Block Diagram*) $\Rightarrow$ Diagrama de Blocos.

Linguagens Textuais

- Lista de Instruções (IL – *Instruction List*) $\Rightarrow$ mnemônicos booleanos;
- Texto Estruturado (ST – *Structured Text*) $\Rightarrow$ parâmetros idiomáticos.

A Figura 2.14 apresenta um exemplo típico utilizando todas as linguagens citadas acima. O exemplo consiste na representação para a expressão lógica $O_1 = (I_0 + I_1) \cdot I_2$ a qual significa que: a saída O_1 será ligada se uma das entradas I_0 ou I_1 for acionada e se I_2 também for acionada. Os exemplos mostrados na Figura 2.14 utilizam linguagens diferentes para realizar a mesma tarefa, embora na programação em SFC o exemplo é apenas ilustrativo pois o resultado da aplicação pode ser relativamente diferente dependendo da sequência em que forem acionadas as entradas I_0 , I_1 ou I_2 .

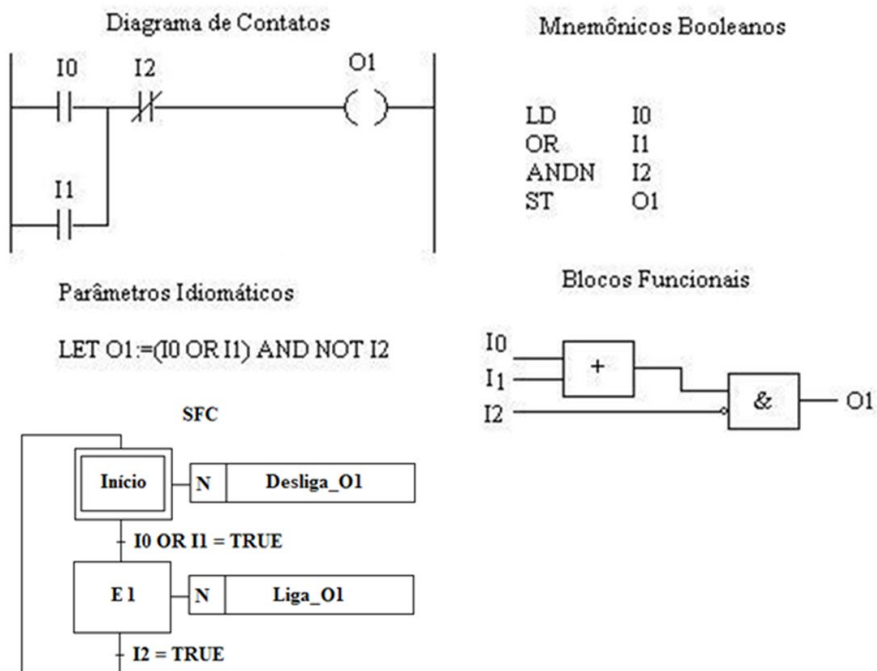


Figura 2.14 – Exemplo de programação utilizando as linguagens previstas na Norma IEC-61131

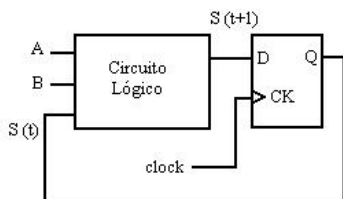
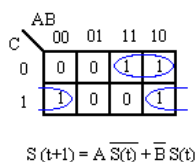
2.6 – Ferramentas para o Desenvolvimento de Aplicações

Na modelagem e controle de sistemas a eventos discretos podem ser utilizadas várias ferramentas gráficas tais como mostradas na Tabela 2.1 e Figuras 2.15 a 2.18. Estas figuras ilustram a utilização de ferramentas diferentes no desenvolvimento da lógica para a mesma aplicação.

- Tabela Verdade (Tabela 2.1) – Descreve o comportamento das saídas discretas do sistema como resultado da combinação lógica das suas entradas. É utilizado na representação de operações combinatórias. Sua utilização combinada com ferramentas como o Mapa de Karnaugh facilita a implementação de aplicações lógicas, resultando em programas e/ou circuitos lógicos simplificados (Figura 2.15).

A(t)	B(t)	S(t)	S(t+1)
0	X	0	0
1	X	0	1
X	0	1	1
X	1	1	0

Tabela 2.1 – Tabela Verdade



Programa em Ladder

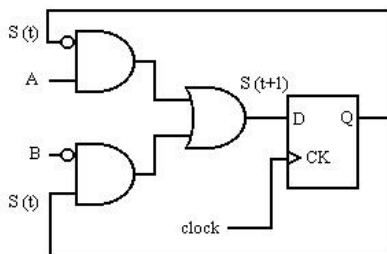
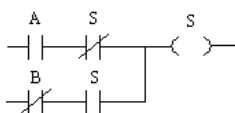


Figura 2.15 – Utilização do Mapa de Karnaugh na Obtenção de Circuito Lógico, Expressão Booleana e Programa em Ladder Simplificados

- Redes de Petri – Modelos gráficos para representação de sistemas discretos constituídos de redes de conexões condição-evento que permite modelar e analisar o comportamento de sistemas sequenciais, tais como o sistema de manufatura flexível.
- Fluxogramas (Figura 2.16) – Ferramenta consagrada em programação textual e linguagem de alto nível (Exemplo: STL, assembly, C), mas também uma excelente ferramenta para programação gráfica, principalmente quando a aplicação é relativamente complexa e apresenta funcionamento de forma sequencial.

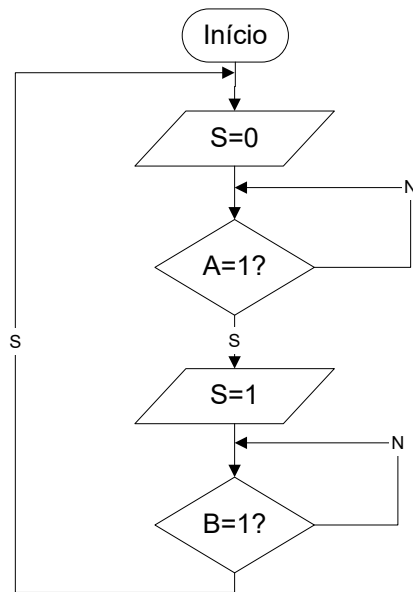


Figura 2.16 – Fluxograma

- Diagramas de variáveis de estado (Figura 2.17) - Descrevem a dinâmica sequencial em dispositivos controladores (Exemplo: máquina de estado, programação de microcontroladores).

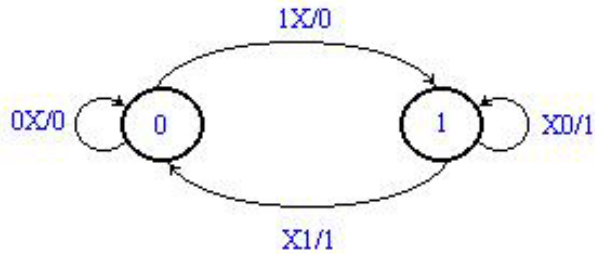


Figura 2.17 – Variáveis de Estado

- Diagrama Trajeto-Passo (Figura 2.18) – Representação gráfica aplicada tipicamente em aplicações de automação utilizando tecnologia pneumática. Embora limitada, por não apresentar visão geral do processo em nem compromisso com tempo, mostra de forma clara a execução sequencial de dispositivos na pneumática.

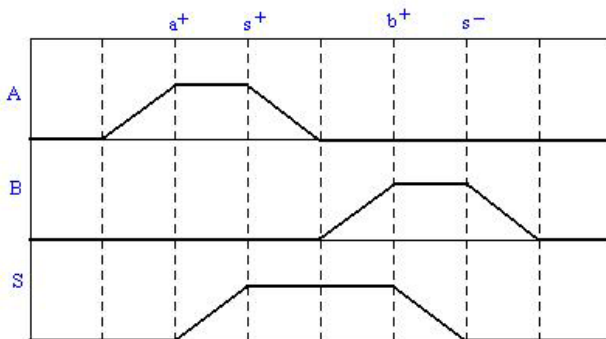


Figura 2.18 – Diagrama Trajeto-Passo

No desenvolvimento dos projetos em laboratório sugere-se utilizar adequadamente estas ferramentas, pois elas facilitam consideravelmente a programação.

A utilização das ferramentas acima e seu emprego no projeto de sistemas lógicos e sequenciais são apresentados em literaturas específicas. Particularmente as ferramentas Tabela Verdade, Mapas de Karnaugh e Diagrama de Variáveis de Estado (ou Máquinas de Estado) são tratadas na maioria dos livros de Eletrônica Digital e consiste na base do conhecimento essencial necessária para a adequada compreensão das aplicações com controladores lógicos programáveis. Assim, esses conceitos não serão detalhados nesse trabalho por considerar que esses conhecimentos já devem ter sido adquiridos pelos leitores. Alguns exemplos de literaturas que abordam esses conceitos que podem ser citados: IDOETA e CAPUANO (1997), UYEMURA (2002), TOCCI e WIDMER (2003), CAPUANO (1991), ERCEGOVAC et. al (2000), FREGNI e SARAIVA (1995), MANO e KIME (2000) e TAUB e SCHILLING (1982).

Fluxogramas são ferramentas essenciais no ensino de programação e são abordadas em diversas literaturas relacionadas a algoritmos e técnicas de programação (FARRER, 1999) (SOUZA et. al, 2011).

Diagramas Trajeto-Passo são ferramentas especificamente voltadas para aplicações de pneumática e hidráulica (FIALHO, 2003) (BONACORSO e NOLL, 1999).

Redes de Petri são abordadas em literaturas mais especializadas voltadas para a modelagem de sistemas a eventos discretos, tais como MIYAGI (1996) e MORAES e CASTRUCCI (2007).

3 – PROGRAMAÇÃO EM LADDER

A programação em ladder utiliza os seguintes tipos de operandos:

- Operandos digitais – Podem assumir as condições ligado ou desligado (0 ou 1). Podem ser de dois tipos: auxiliares internos (posições de memória) e pontos físicos (entradas e saídas digitais).
- Operandos analógicos – Representam valores de temporização, contagem, *set-point*, sensores analógicos, etc.

Instruções Básicas

Para a programação em ladder utilizam-se instruções, onde cada instrução ocupa uma célula lógica. Algumas instruções básicas da programação em ladder são ilustradas na Figura 3.1.

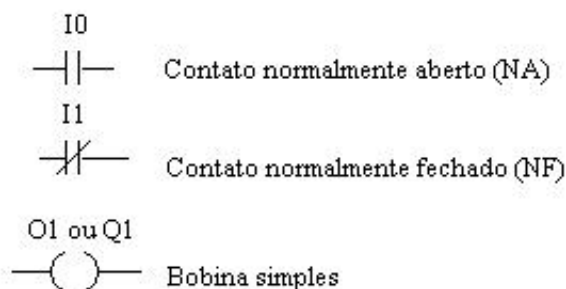


Figura 3.1 – Comandos básicos da programação em ladder

Os contatos NA e NF podem ser usados para monitorar entradas, saídas ou auxiliares internos digitais. As bobinas são utilizadas para ligar/desligar saídas ou auxiliares internos (bobinas auxiliares).

O diagrama de contatos de um programa realizado em linguagem ladder consiste em um desenho formado por duas linhas verticais, que representam os polos positivo e negativo de uma bateria ou fonte de alimentação genérica. Entre essas duas linhas verticais são desenhados ramos horizontais que possuem chaves, que podem ser de lógica normalmente aberta ou fechada, e que representam os estados das entradas do CLP. Nesses ramos horizontais são representadas as saídas do CLP (bobinas), de maneira que o estado delas depende do estado das entradas desse mesmo ramo. Alguns exemplos básicos são apresentados na Figura 3.2 que ilustra a representação da linguagem ladder para a implementação de funções lógicas básicas.

Os softwares de programação de CLPs em geral permitem identificar os pontos de entrada e saída do diagrama ladder através de *label* (TAG) através do qual o usuário pode identificar mais claramente os pontos de entrada. O usuário pode utilizar o endereço do operando (I0, I1, I2, O0, O1, Q0, Q1, M0, M1, F0, F1,...) ou TAG (sensor0, solenoide1, chave2, aux1,...) para programar.

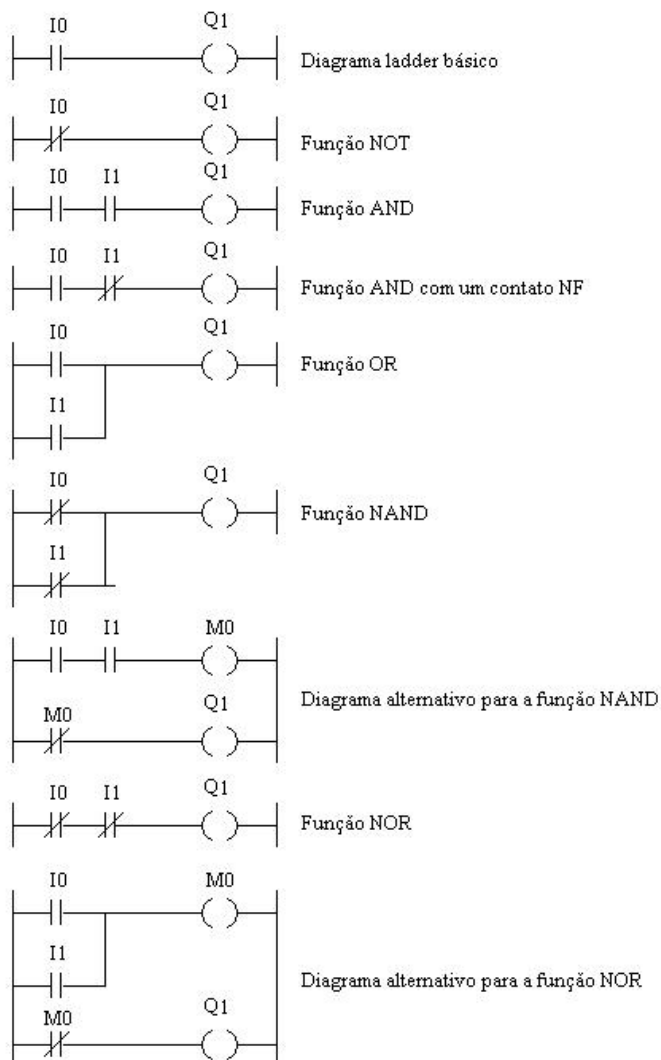


Figura 3.2 – Programas em ladder para representação de funções lógicas básicas

A estrutura básica da programação em ladder é montada em um espaço de 8 colunas equivalente a 32 células, para a criação do programa ladder. Um programa é formado por várias lógicas. Uma lógica pode ter menos que 4 linhas, porém precisa conter pelo menos 1 instrução. A Figura 3.3 apresenta um programa com duas lógicas, sendo cada lógica constituída de 32 células (4 linhas e 8 colunas).

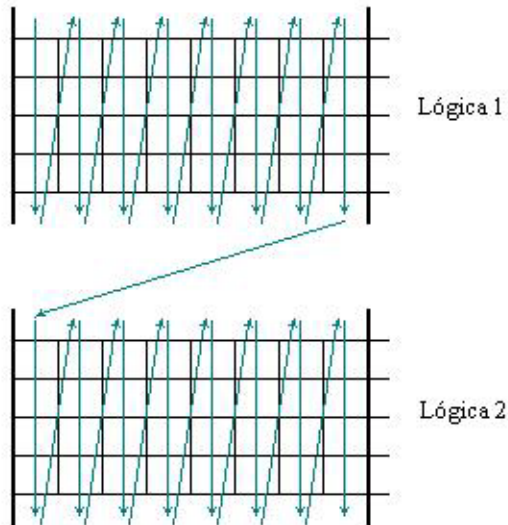


Figura 3.3 – Varredura em duas lógicas consecutivas

Alguns CLPs não possuem limites para o número de linhas por lógica, mas isto em nada interfere nos conceitos de programação, uma vez que o programador pode continuar seu programa na próxima lógica. O número limite de lógicas dependerá do tamanho da memória disponível. Em uma lógica a varredura das instruções é feita de cima para baixo e da esquerda para a direita. Na Figura 3.3, o sentido das varreduras ao longo das duas lógicas é apresentado.

Para entender o funcionamento da lógica e verificar o que acontece em cada varredura considere o seguinte exemplo ilustrado na

Figura 3.4, onde utilizamos 5 entradas, uma bobina auxiliar A_0 e a saída S_1 . Fica claro que a saída S_1 será o resultado da lógica AND entre as 5 entradas. Contudo, se todas as entradas estiverem ligadas e considerando a ordem em que as células são “varridas” pelo processador (de cima para baixo e da esquerda para a direita) pode-se observar que nesta solução a saída S_1 demorará 2 varreduras para ligar. Na primeira varredura o contato auxiliar A_0 está desligado, e só vai ligar após a bobina A_0 ligar.

Na Figura 3.5, temos um exemplo similar, mas utilizando 2 lógicas. Neste caso a saída S_1 liga na primeira varredura. Aparentemente no exemplo com 2 lógicas S_1 liga mais rapidamente porque é necessário somente um ciclo de varredura. Contudo, considerando que cada ciclo de varredura é executado em microssegundos, isto fica praticamente imperceptível. Além disso, nesta situação, é necessário que durante o processamento em cada lógica sejam efetuadas leituras em todas as entradas, saídas e auxiliares envolvidos o que faz com que o tempo de varredura do programa seja maior que no exemplo anterior com somente 1 lógica.

É importante destacar que, nem todos os fabricantes apresentam software que permitem programação em lógica separada. Também em muitos programas a varredura não é efetuada na vertical, como ilustrado na Figura 3.4. Ao contrário, em alguns modelos a varredura é efetuada na horizontal ou até mesmo na diagonal.

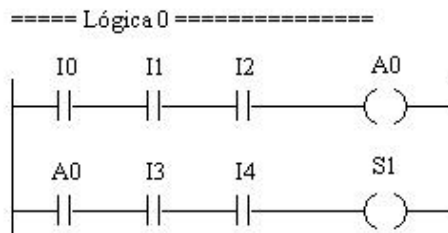


Figura 3.4 – Uma aplicação constituída somente de uma lógica

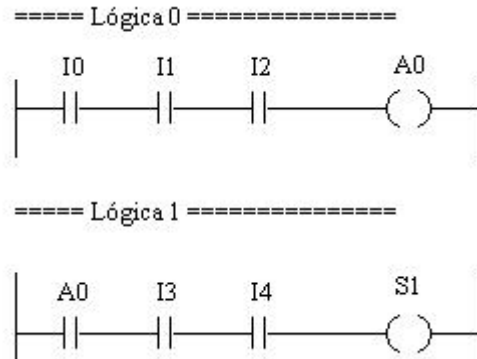


Figura 3.5 – Uma aplicação constituída de duas lógicas separadas

Circuitos de Intertravamento

Até agora todos os exemplos mostrados respondem à lógica combinacional. Agora consideraremos aplicações onde o estado das saídas depende das entradas, mas também da condição inicial das saídas (lógica sequencial).

Por exemplo, considere o diagrama a seguir (Figura 3.6) onde se I_0 e I_1 desligadas, o estado do auxiliar M_0 e, portanto, da saída Q_0 dependem do estado anterior de M_0 . Pode-se observar que, neste caso, somente um pulso em I_0 é suficiente para manter a saída Q_0 ligada. Contudo, estando M_0 ligada (e, portanto, Q_0 também ligada), a única maneira de desligá-la é ligando I_1 . Em síntese, I_0 funciona como um pulsador de *start* e I_1 funciona como um pulsador de *stop*.

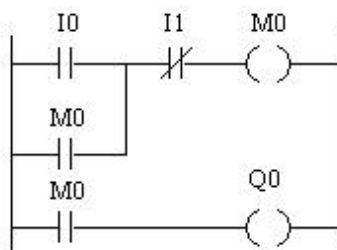


Figura 3.6 – Circuito de intertravamento com selo

Finalmente, cabe destacar que alguns modelos de CLP permitem utilizar uma saída real também como entrada virtual. Nesse caso, a utilização da variável M_0 é desnecessária.

Sentenças SET e RESET

Uma outra maneira de conseguir um intertravamento é através do uso da sentença SET, a qual permite ativar uma saída ante uma condição de entrada e permanecer nesta condição mesmo mudando o estado da entrada. Para desativar a saída, deve-se utilizar a sentença RESET, a qual desativa a saída ante uma condição da entrada e permanece desativada mesmo mudando essa condição de entrada.

O diagrama abaixo (Figura 3.7) apresenta um exemplo de circuito de intertravamento, equivalente ao demonstrado anteriormente, mas usando as bobinas de SET e RESET.

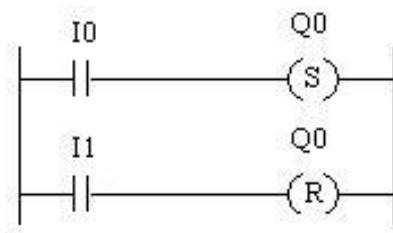


Figura 3.7 – Utilização de bobinas retentivas SET e RESET

Circuitos de Detecção de Borda

Existem algumas situações em que é necessário registrar não o estado de uma entrada, mas o instante em que essa entrada comuta. Isso é realizado por meio de circuitos de detecção de borda (Figura 3.8), que podem detectar tanto o flanco ascendente como o descendente. Isto pode ser conseguido através do seguinte diagrama:

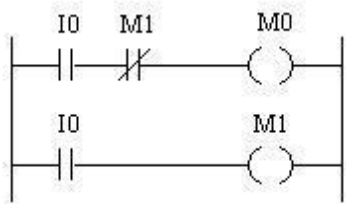


Figura 3.8 – Circuito de detecção de borda

Considere inicialmente que a entrada I_0 e as variáveis M_0 e M_1 estão em estado lógico 0. Quando I_0 é ligado, no primeiro ciclo de varredura colocará $M_0 = M_1 = 1$. Mas no seguinte ciclo de varredura, ao ter lido um 1 na variável M_1 , colocará um 0 na variável M_0 . Isso implica que a variável M_0 permanecerá no estado lógico 1 apenas durante um ciclo de varredura. O estado dessa variável será um impulso de curta duração que aparece no instante de ativação da entrada I_0 . O diagrama de tempos da Figura 3.9 representa essa situação.

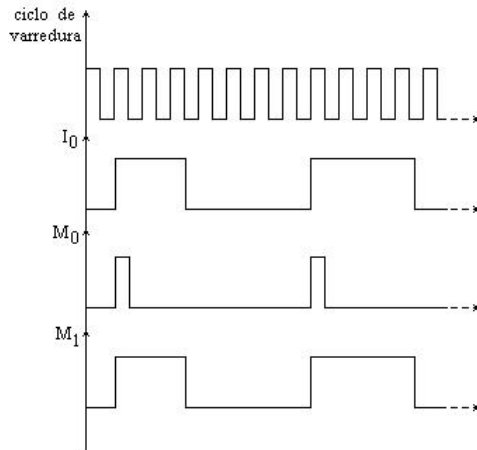


Figura 3.9 – Formas de onda obtidas no circuito de detecção de borda

Um exemplo de aplicação prática desse circuito é quando se deseja ativar e desativar uma saída com um único pulsador. Quando este é pressionado, se ativará uma saída, e quando for pressionado pela segunda vez, será desativada, e assim por diante. O diagrama completo para essa implementação está ilustrado na Figura 3.10.

No diagrama, os dois primeiros ramais são iguais ao do diagrama anterior onde M_0 só se ativará na borda de subida de I_0 e permanecerá ativada durante um impulso. Ao se ativar M_0 , ativar-se-á M_2 , através da terceira linha (ramal) do diagrama, acionando, portanto, a saída Q_0 . Um ciclo de varredura depois, ao se desativar M_0 , o terceiro ramo ainda continuará fechado, devido a que a linha inferior se fechará agora, mantendo M_2 ativado e, por conseguinte, a saída Q_0 também. Ao se ativar pela segunda vez I_0 , será ativada a variável M_0 abrindo a linha inferior do terceiro ramo e, logo, desativando M_2 e consequentemente a saída Q_0 . No seguinte ciclo de varredura, ao se desativar a variável M_0 , a variável M_2 ainda permanecerá desativada, por estar o terceiro ramo aberto, prosseguindo a saída Q_0 desativada.

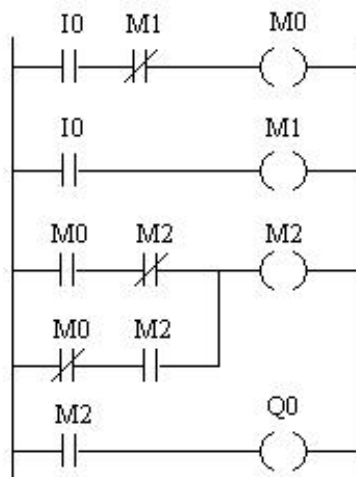


Figura 3.10 – Uma aplicação utilizando entrada pulsante que permite uma mesma entrada executar ações de ligar e desligar usando circuito de detecção de borda

Uma outra forma de desenvolver a detecção de borda para controlar o ligamento e desligamento utilizando uma mesma entrada é ilustrado na Figura 3.11, neste caso utilizando bobinas retentivas (Set e Reset). Note que as quatro linhas do programa são utilizadas respectivamente para detectar:

- o primeiro acionamento em I_0 (por exemplo, para ligar a saída Q_0);
- o primeiro desacionamento em I_0 ;
- o segundo acionamento em I_0 (por exemplo, para desligar a saída Q_0);
- o segundo desacionamento em I_0 .

Note também que o estado da saída Q_0 é invertido toda vez que a entrada pulsante I_0 é acionada.

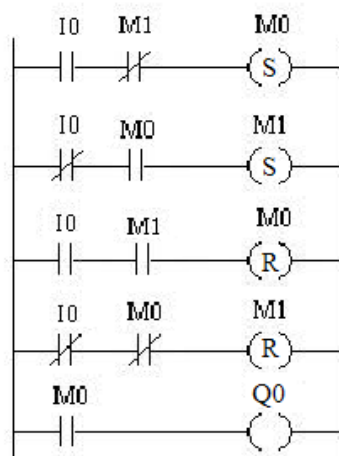


Figura 3.11 – Uma aplicação com circuito de detecção de borda controlado por bobinas retentivas

Temporizadores

A maioria dos CLPs atuais possui temporizadores internos. A maneira como ele é inserido no diagrama de contatos, assim como algumas particularidades sobre seu uso são próprias de cada modelo de CLP. Daremos aqui uma definição genérica, a qual deve ser adaptada segundo o manual fornecido pelo fabricante na hora de ser implementado um circuito com temporizador.

O diagrama básico de um temporizador é apresentado na Figura 3.12 onde I_0 é a entrada que controla a ativação do temporizador, TMP (*Timer Preset*) é o valor de ajuste do temporizador e TB (*Time Base*) é a base de tempo determinada. A base de tempo em geral pode ser escolhida entre algumas opções pré-determinadas.

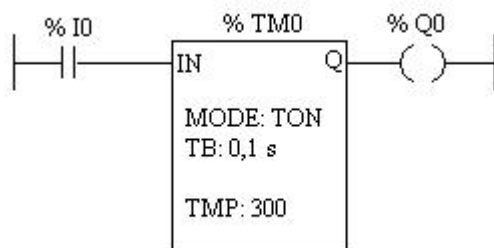


Figura 3.12 – Estrutura básica de linha de programação de um temporizador em ladder

Nesse exemplo, a base de tempo escolhida é 0,1 s e $TMP = 300$, o que significa que o tempo de retardo deste temporizador equivale a 30 s. O símbolo % especifica endereço de memória, que pode ser de por exemplo entrada (%I0), saída (%Q0), auxiliar (%M0) ou temporizador (%TM0). Em alguns CLPs o símbolo % não é necessário.

Para entender o funcionamento deste temporizador (em alguns CLPs conhecidos como TON), analisemos o diagrama de tempos abaixo onde podemos perceber que, quando se ativar a entrada I_0 , começará uma contagem de impulsos na base de tempo especificada. Decorrido o tempo de 30 s (300 impulsos) será ativada a saída Q_0 . Se, no entanto, a entrada I_0 se desativar antes da contagem finalizar, a saída Q_0 não se ativará em momento algum, zerando a contagem armazenada em um temporizador / registrador interno (TM_0).

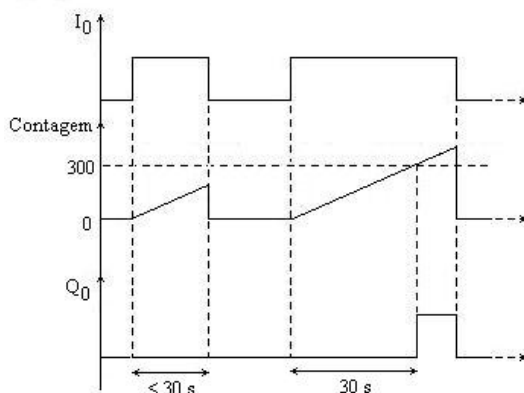


Figura 3.13 – Formas de onda em uma aplicação com temporizador

Além desse diagrama básico, é possível implementar outros diagramas de contatos que utilizam temporizadores e apresentam outros efeitos. A título de ilustração, serão apresentados dois exemplos.

O primeiro exemplo (Figura 3.14) consiste em ativar uma saída quando se ativa uma entrada e mantê-la nessa condição até 30s depois da entrada se desativar (efeito de retardo no desligamento). O diagrama de contatos que produz tal efeito e o diagrama de tempos correspondente são mostrados a seguir.

Cap. 3 – Programação em Ladder

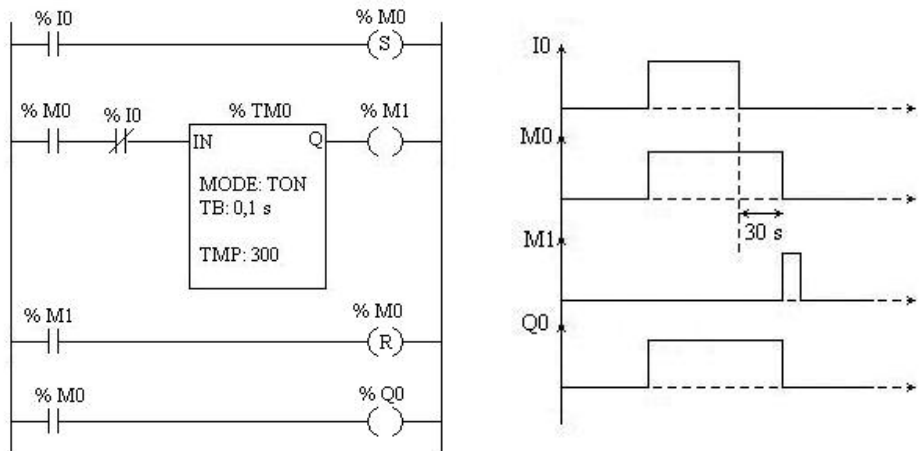


Figura 3.14 – Uma aplicação com temporizador

O segundo exemplo (Figura 3.15) consiste na implementação do clássico “oscilador astável”, realizado com dois temporizadores. O diagrama de contatos e o diagrama de tempos correspondente são mostrados a seguir.

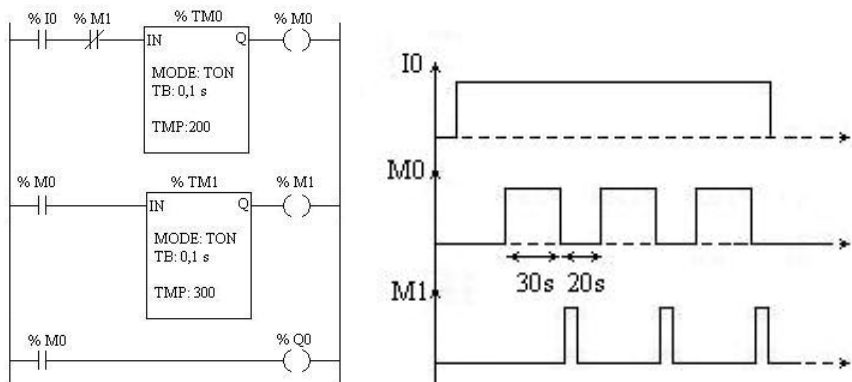


Figura 3.15 – Oscilador astável construído por temporizadores

Um outro tipo de temporizador (conhecido geralmente como TOFF) apresenta a contagem de tempo zerada enquanto a entrada de controle estiver ativada. Diante da desativação da entrada de controle, inicia-se a contagem de tempo, a qual se detém quando chega ao tempo ajustado (valor presetado vezes a base de tempo). A saída prossegue ativada sempre que a contagem de tempo for menor que o tempo ajustado, o que inclui o tempo que a entrada de controle continuar ativada (já que durante esse tempo a contagem é igual a zero). O exemplo da Figura 3.16 ilustra a situação.

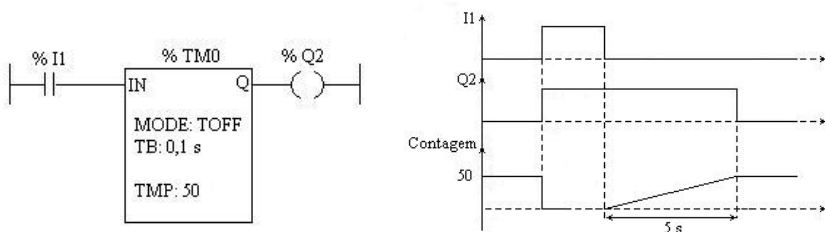


Figura 3.16 – Uma aplicação básica com temporizador TOFF e formas de onda resultantes

Em geral os CLP's disponibilizam ainda outros tipos de temporizadores. Contudo as características de funcionamento e particularidades da configuração de cada modelo são tão diferentes que estes não serão apresentados, devendo o usuário consultar os manuais quando da utilização destes temporizadores.

Contadores

Um outro recurso muito útil que a maioria dos CLPs traz são os contadores, os quais permitem quantificar a ocorrência de eventos e executar uma determinada ação depois de um determinado número de ocorrências. A maneira de implementar um contador em um diagrama ladder também varia bastante de um modelo de CLP para outro, devendo o usuário se informar sobre a forma de implementação específica no programa utilizado. Alguns CLPs apresentam também diferentes tipos de contadores: crescente, decrescente ou em alguns casos com uma

entrada de contagem crescente e outra decrescente no mesmo contador. Daremos aqui um exemplo genérico.

O contador tem a função de contar o número de ativações (bordas de subida) da entrada de controle. Possui duas entradas de controle para contagem crescente (CU – *Counter Upper*) e decrescente (CD – *Counter Down*), uma entrada de reset (R), uma entrada de set (S) e ajuste do valor de contagem CP (*Counter Preset*). Um exemplo ilustra a implementação do contador crescente (Figura 3.17). Neste caso devemos ter $CD = 0$ e $S = 0$. No exemplo, ilustrado abaixo, o valor de contagem foi ajustado em CP = 5. Isto significa que, a saída Q_1 será ativada a partir da quinta ativação da entrada I_1 . O valor instantâneo da contagem é guardado em um registrador interno, sendo zerado através da ativação da entrada I_2 .

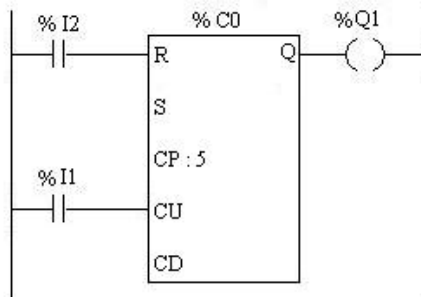


Figura 3.17 – Uma aplicação básica com contador

Outras Instruções

Além dessas instruções, também encontramos disponíveis outras funções tais como operações matemáticas e lógicas, comparação de dados (menor que, igual a, etc.) e de manipulação de dados (deslocamento de bytes e words, registros de deslocamento, conversão entre dados BCD e inteiro, pulos e subrotinas). Cada CLP possui uma maneira específica de implementar essas instruções, sendo, portanto, recomendável consultar o manual de usuário do programa específico do CLP.

Algumas destas funções serão apresentadas nos próximos capítulos onde abordaremos o desenvolvimento de aplicações utilizando três modelos diferentes de CLPs:

- CLP da Omron;
- CLP da Schneider Electric;
- CLP FEC® da Festo.

A abordagem nesses capítulos é em alguns pontos direcionada para a utilização de softwares e modelos específicos. Contudo, o leitor deve considerar que nosso objetivo nesse trabalho não é direcioná-lo para a utilização de um software ou ferramenta específica. Além disso, sabemos que os softwares são atualizados constantemente. O objetivo dessa abordagem é demonstrar diferentes formas de introduzir funções e aplicações por meio da programação dos CLPs.

4 – PROGRAMAÇÃO EM LADDER NO CLP OMRON

Neste capítulo serão apresentados os aspectos de programação do CLP Omron. Daremos ênfase à programação em Ladder do CLP utilizando o Software CX-One. Algumas aplicações típicas serão analisadas sendo desenvolvido o programa específico de implementação. Também neste capítulo serão apresentados aspectos relativos às ferramentas que podem ser utilizadas no desenvolvimento das aplicações.

4.1 – Controlador Lógico Programável CJ2M - Omron.

O CLP CJM2M - CPU31 da OMRON é um controlador modular expansível que pertence à série CJ2, que permite até 5 K *steps* de programação e até 2560 pontos de entrada e saída (I/O), o que equivale a 40 *racks*, além de apresentar 1 porta Ethernet / IP, 1 porta USB e 1 porta serial opcional (CP1W - CIF01).

O equipamento do laboratório (Figuras 4.1 e 4.2) inclui módulo DeviceNet Master (CJ1W - DRM21), módulo de comunicação Profibus - DP Master (CJ1W - PRM21) e permite futura conexão de módulos adicionais, como por exemplo, CanOpen ou Modbus-RTU, além de outros dispositivos, como sistema de visão, controladores de processo, sensores inteligentes, etc.

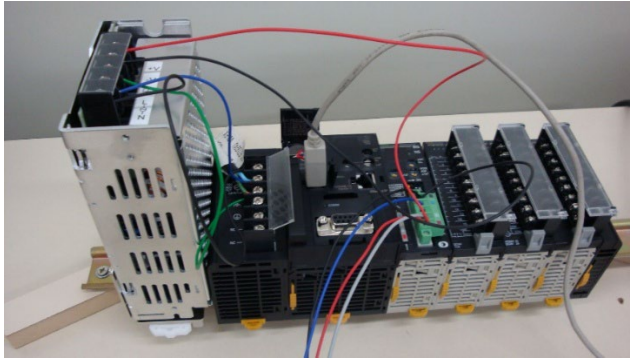


Figura 4.1 – CLP Modular OMRON incluindo módulos I/O e módulos de rede

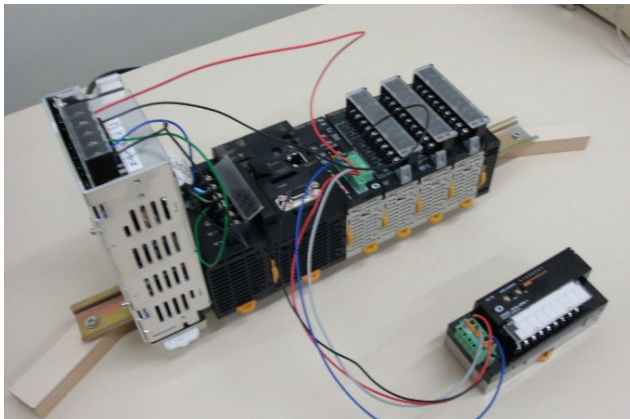


Figura 4.2 – CLP Modular OMRON conectado a um módulo remoto DeviceNet

Encontram-se disponíveis fonte de alimentação (CJ1W - PA202), módulo de entrada digital de 16 pontos (CJ1W - ID211), módulo de saída digital a relé de 16 pontos (CJ1W - OC211), módulo misto com 4 entradas e 2 saídas analógica (CJ1W - MAD42), módulo remoto DeviceNet com 8 pontos de entrada digital PNP (DRT2 - ID08-01), módulo remoto DeviceNet

com 8 pontos de saída digital PNP (DRT2 - OD08-01), fonte de alimentação externa 24 VCC e 4 A para módulos de rede, interface homem-máquina (IHM) colorida com tela de 5,6", módulo Profibus DP Slave para Inversor de Frequência e Inversor de Frequência vetorial de 220 V e 0,4 kW (0,5 CV).

O modelo permite a realização de operações com reduzido ciclo de varredura. Num pior caso, considerando uma aplicação com 5K linhas de programação com ponto flutuante, o CLP consegue operar em aproximadamente 0,4 ms de tempo de varredura.

4.2 – Instruções para Programação Utilizando o Software CX-One.

A programação do CLP CJ2M - Omron é efetuada diretamente através do software CX-One instalado num PC (*Personal Computer*).

O software CX-One é um pacote de programação integrado que permite realizar a programação do CLP da OMRON utilizando as linguagens Ladder, Lista de Instruções e SFC. Adicionalmente o software permite realizar simulações, configurações de rede e programar a IHM. O pacote inclui o CX-Server, que permite comunicação com servidor.

Esse programa requer licença para utilização. Uma versão para estudante, dedicada à programação do CLP modelo CP1E encontra-se disponível na página da OMRON (<http://www.omron.com.au>). Com essa versão é possível elaborar programas, transferir para um CLP virtual e depois realizar simulações ou converter para testar no equipamento real.

Seguem algumas instruções básicas para a programação e conexão de hardware.

- A comunicação com PC é feita pela porta USB, ou se preferir, utilizando rede Ethernet.
- O software é acessado pelo menu Iniciar (*Start*) > OMRON > CX-One > CX-Programmer > CX-Programmer.exe. A janela de inicialização do software (Figura 4.3) é apresentada.

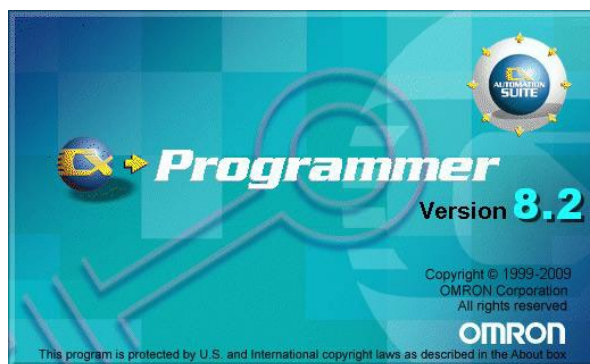


Figura 4.3 – Janela de inicialização do programa CX - Programmer

- Clique em *New Project*, escolha um nome para o projeto, selecione o modelo do CLP utilizado (*Device Type*) e em *Settings* o modelo do processador (CPU31), e conforme ilustrado na Figura 4.8. O modo de comunicação (*Network Type*) é USB. Caso a rede Ethernet / IP já tenha sido montada para permitir a comunicação entre a CPU e o PC, então ela pode ser selecionada.

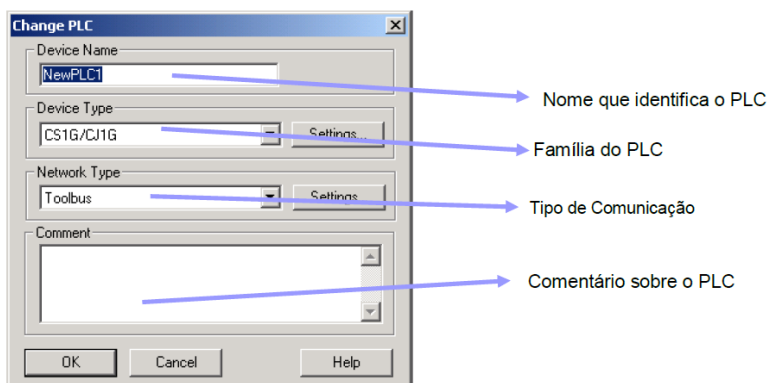


Figura 4.4 – Seleção do CLP que será utilizado

Identificação dos Dispositivos

- No próximo passo ao clicar em *I/O Table and Unit Setup*, aparecerá uma janela contendo a tabela com todos os cartões e racks que estão sendo utilizados no PLC, e seus respectivos endereços de memória na CPU (Figura 4.9).
- A posição dos módulos de IO deve ser mantida, caso contrário poderá ser necessário alterar programas anteriormente elaborados. Isso ocorre porque os IO's não têm endereço físico configurável, tal como ocorre com os módulos de rede. Assim esses endereços são alocados automaticamente dependendo da disposição no rack. Para os módulos de rede, ao contrário, o endereço é definido pelo ajuste em Unit N., no próprio módulo.

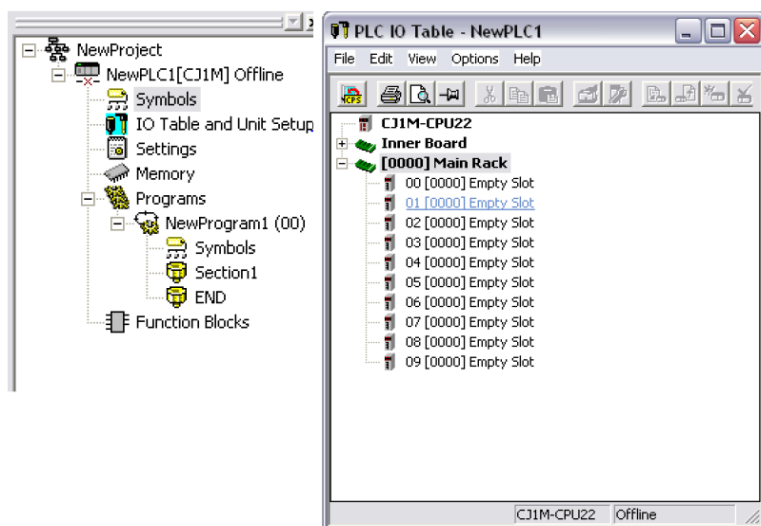


Figura 4.5 – Criação do IO Table

- Dependendo da ação que se deseja realizar é necessário definir por software o modo de operação. As opções disponíveis dependem do

CX-Programmer estar em Modo *Off-line* ou *Online* e se o CLP está em Modo *Program* ou *Monitor/Run*. Certifique-se que o CLP está em Modo *Program*, e na janela do *I/O Table* clique em *Options > Create* e automaticamente o *software* criará a tabela para você.

- É muito importante o entendimento detalhado do modo de funcionamento. Pois existe diferença no tipo de funcionamento para descarregar programa, descarregar novas configurações, alterar programas ou mesmo deixar em modo de execução. A Figura 4.6 mostra as barras de identificação do modo de funcionamento.

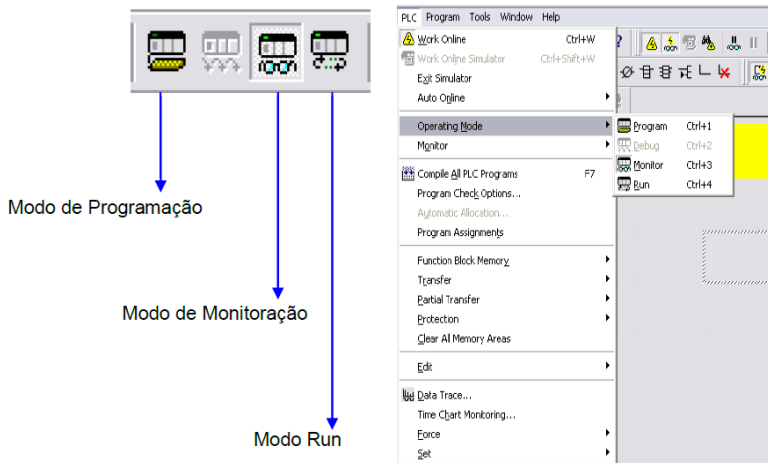


Figura 4.6 – Modos de Operação

- **Modo de Programação:** Neste modo pode-se forçar valores a qualquer área de memória, fazer edição on-line do programa e somente neste modo pode-se transferir dados para o CLP, porém nenhuma instrução será executada e os bits só serão acionados se forçados.

- **Modo de Monitoração** (CTRL M): Neste modo pode se forçar valores a qualquer área de memória, fazer edição *on-line* do programa, e todas as instruções são executadas e os bits acionados conforme a lógica programada. Para simular os acionamentos basta selecionar o contato e clicando com o botão direito do *mouse* podemos forçar os estados desejados. Atalhos também podem ser utilizados para ligar (CTRL J), desligar (CTRL K) e cancelar um *force* (CTRL L).

- **Modo Run** (CTRL W): Neste modo não é permitido se forçar nenhum valor, nem fazer edição *On-line* do programa, e todas as instruções são executadas e os bits acionados conforme a programação.

Criação de *Tags*

- Ainda antes de realizar a programação é possível criar *tags*. *Tags* são objetos do sistema os quais são associados a variáveis para permitir seu controle por meio de um protocolo ou rede de comunicação, por exemplo, utilizando um supervisor, tal como o software Eclipse SCADA ou o IWS (Indusoft Web Studio). Nesse caso, para cada objeto inserido na tela, devemos associar pelo menos um *tag*, por exemplo, o nível de um alarme, a temperatura de um forno, uma unidade de memória, uma entrada digital, etc.
- O CLP deve estar no modo *off line* para permitir a criação de *tags*.
- Há várias *tags* já criadas para funções específicas.
- Para criar uma *tag* realize os seguintes passos:
 - clique em *Insert > Symbol*;

Cap. 4 – Programação em Ladder no CLP Omron

- defina o nome; exemplo: *Name*: Entrada 01
- defina o tipo de dado (booleano, inteiro, *word*, etc.); exemplo: *Data type*: BOOL;
- defina o endereço ou valor da variável; exemplos: *Address / Value*: 0.00 (canal ou módulo de entrada / saída e o bit associado).

Exemplos:

a) *Name*: Entrada 01

Data type: BOOL

Address / value: 0.00

Canal de entrada 00

b) *Name*: Saída 01

Data type: BOOL

Address / value: 1.00

Canal de saída 00

c) *Name*: Tempo

Data type: BOOL

Address / value: t0

Temporizador t0

Criação de Programa Simples em Ladder

1º Passo – Identificação de módulos

- Coloque o programa no modo *Work Online*;
- Selecione PLC / Mode Programming;
- Selecione *IO Table / Create*

2º Passo – Programação

- Coloque o programa no modo *Work Offline*;

- Desenvolva o programa em Ladder, utilizando a paleta de edição (Figura 4.11) disponível na barra superior do software de programação CX-Programmer.
 - Coloque o programa no modo *Work Online* novamente;
 - Selecione PLC / Mode Programming;
 - Transfira o programa para o PLC por meio do atalho *PLC / Transfer to PLC*;
 - Coloque o PLC no modo *Run* (*PLC / Mode Run*).
- Agora o PLC já está rodando com o programa gravado.

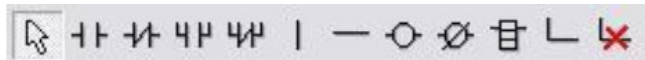


Figura 4.7 – Paleta de Programação em Ladder

A função de cada um dos ícones sequencialmente apresentados na barra de ferramentas pode ser assim resumida:

- *Selection Mode* - permite escolher uma ou mais instruções, editá-las ou apagá-las.
- *New Contact (C)* - permite inserir novo contato NA (Normalmente Aberto).
- *New Closed Contact (/)* - permite inserir novo contato NF (Normalmente Fechado).
- *New Contact OR (W)* - permite inserir função OR com contato NA.
- *New Close Contact OR (X)* - permite inserir função OR com contato NF.
- *New Vertical (Ctrl+Down)* - permite criar ou apagar linha vertical.
- *New Horizontal (Ctrl+Right)* - permite criar ou apagar linha horizontal.
- *New Coil (O)* - permite inserir bobinas NA.
- *New Closed Coil (Q)* - permite inserir bobinas NF.

- *New PLC Instruction (I)* - permite inserir nova instrução como temporizadores, contadores, movimentadores de dados, deslocadores de dados, etc.
- *Line Connect Mode* - permite criar novas linhas na direção vertical ou horizontal.
- *Line Delete Mode* - permite deletar linhas na direção vertical ou horizontal.

A Figura 4.8 ilustra um exemplo de programação utilizando um contato NF para acionar uma bobina NA.

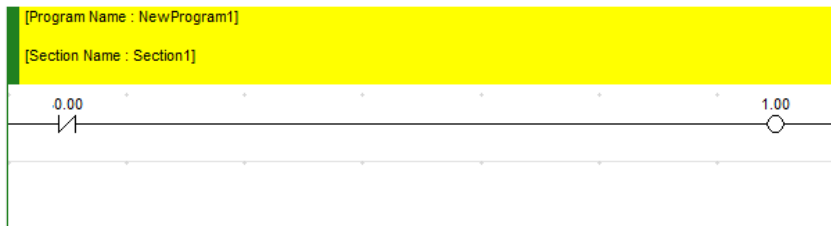


Figura 4.8 – Exemplo Básico Utilizando Contato NF e Bobina NA

A Figura 4.9 mostra a utilização de funções especiais SET e RSET (reset) que podem ser inseridas utilizando o ícone *New PLC Instruction*. Lembre-se que essas funções permitem realizar funções de intertravamento, respectivamente ativando ou desativando a saída quando a entrada da função estiver habilitada. Note que no exemplo da Figura 4.13 as funções Set e Reset são controlados por entradas pulsantes. Assim, basta um pulso na entrada 0.00 para setar a saída 1.00, ou um pulso na entrada 0.01 para resetar a saída 1.00.

Os pulsos utilizados no exemplo são produzidos por uma função especial constituída de uma *tag* já disponível no PLC, o qual pode ser identificada pela nomenclatura P_1s. Essa *tag* resulta em um pulso de 1 segundo que é produzido em toda vez que essa linha for habilitada. Assim, como consequência dessa aplicação, esse exemplo produz um pisca-pisca a 0,5 Hz.

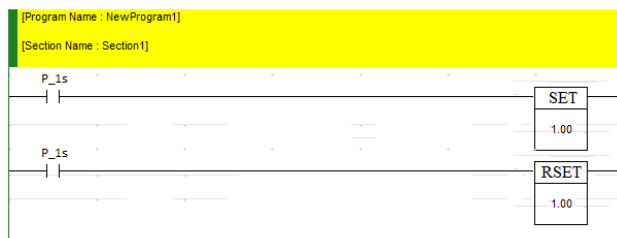


Figura 4.9 – Pisca-Pisca Utilizando Set e Reset

4.3 – Programação de Temporizadores e Contadores.

Para a realização da experiência vamos utilizar o controlador lógico programável da Omron e o software CX-One, nesse caso tornando necessário utilizar temporizadores. O funcionamento dos temporizadores, contadores e outras funções especiais no software CX-One as quais serão detalhadas a seguir. Contudo, antes disso é necessário apresentar informações relativas à divisão de área de memória nos controladores lógicos programáveis da Omron.

Divisão de Área de Memória

A área de memória dos controladores lógicos programáveis, para efeito de programação utilizando o software CX-Programmer é subdividida em várias partes. Essas partes são identificadas na Tabela 4.2.

PREFIXO	DESIGNAÇÃO	SIGNIFICADO
IR	<i>Internal Relay</i>	Relés internos
SR	<i>Special Relay</i>	Relés especiais
HR	<i>Hold Relay</i>	Relés de retenção
AR	<i>Auxiliary Relay</i>	Relés auxiliares
LR	<i>Link Relay</i>	Relés de ligação
TC	<i>TIM / CNT</i>	Temporizadores / Contadores
DM	<i>Data Memory</i>	Memória de Dados

Tabela 4.2 – Divisão de Área de Memória

Para verificar o conteúdo bem como o espaço de endereçamento de cada uma dessas partes da memória basta selecionar a opção CIO da CPU CJ2M na árvore de funções do software, como mostrado na Figura 4.10.

A função de cada uma dessas partes da memória pode ser assim resumida:

Área Interna (IR) - Essa área compreende os canais associados aos terminais físicos externos (entradas e saídas), mas também podem ser utilizados como relés de entrada / saída, independentes de sua conexão com o meio externo. Apresentam estados não retentivos, ou seja, não retêm o estado caso ocorra queda na alimentação ou mudança no modo de operação.

Área Especial (SR) - São relés de sinalização associados a funções especiais para diagnóstico, indicação de operação (ON / OFF), flags de sinalização de resultados de cálculos (>, <, =), comunicações, temporizações, etc.

Área Auxiliar (AR) - Contém bits de controle e informação de recursos do PLC como, porta RS232C, porta periférica, cartão de memória, etc., não disponíveis para o usuário. Dividem-se em duas partes: sinalização (erros de configuração, dados do sistema) ou memorização e gestão de dados.

Área de Ligação (LR) - Contém bits utilizados no intercâmbio de dados entre dois PLC's interligados por computador. Caso uma interligação não seja necessária, podem ser utilizados como bits de trabalho normalmente.

Área de Retenção (HR) - Espaço de memória reservado para utilização em programação convencional, com a vantagem de manter o seu estado em caso de queda na alimentação ou alteração no modo de operação. Nesse caso utiliza-se bateria de lítio que permite manter a informação memorizada mesmo quando houver queda de energia. Podem

ser manipulados normalmente, como os IR`s, sendo endereçados como bit ou como canal.

Memória de Dados (DM) - Tratam-se de memórias de 16 bits (*word's*) utilizados para gerenciar valores numéricos. Mantém o seu estado com as alterações no modo de trabalho ou em caso de queda na alimentação.

Temporizadores e Contadores (TIM e CNT) - Área de memória que simula o funcionamento de temporizadores e contadores. São usados pelos PLC's para armazenar informações como valor de pré-seleção (SV), valor atual (PV), bit valor do estado, etc. Trabalhando no modo on-line, monitor, é possível acessar esse espaço de memória para visualizar os estados dos temporizadores, por exemplo, e identificar qual deles está ligado.

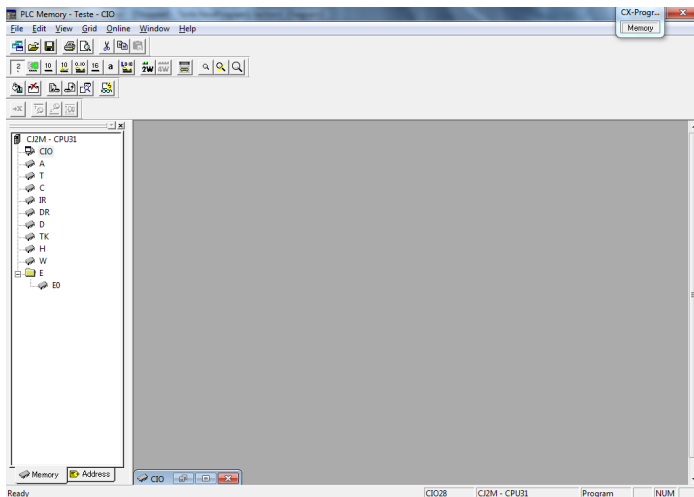


Figura 4.10 - Visualização das Divisões da Área de Memória no Software CX - Programmer

Considerando aspectos de programação para aplicações básicas, utilizando entradas, saídas e memórias auxiliares, é importante destacar que:

- entradas e saídas são armazenadas na área interna (IR), e são endereçados conforme a disposição dos módulos, sequencialmente após a CPU; por exemplo, se colocamos um canal de entrada após a CPU, então essa será indexada pelo índice 0, e cada canal de entrada usará endereços referenciados por 0.00, 0.01, 0.02, e assim por diante; se na sequência introduzimos um módulo de saída, esse será indexado pelo índice 1, e cada canal de saída usará endereços referenciados por 1.00, 1.01, 1.02, e assim por diante; dispositivos de comunicação em rede, módulos de entrada e saída analógica além de módulos especiais não obedecem a esse critério e seu endereço de acesso tipicamente depende da configuração dos módulos, fisicamente, utilizando os ajustes Unit nº.;
- memórias auxiliares não-retentivas utilizadas na programação podem ser armazenadas na área interna (IR); nesse caso as memórias não-retentivas devem obedecer ao padrão de identificação w0.00, w0.01, w0.02, e assim por diante;
- memórias auxiliares retentivas utilizadas na programação podem ser armazenadas na área de retenção (HR); nesse caso as memórias retentivas devem obedecer ao padrão de identificação h0.00, h0.01, h0.02, e assim por diante.

Funções Especiais

Para introduzir funções especiais na programação em Ladder do software CX-Programmer deve-se utilizar o ícone *New PLC Instruction* disponível na paleta de edição como mostrado na Figura 4.11.

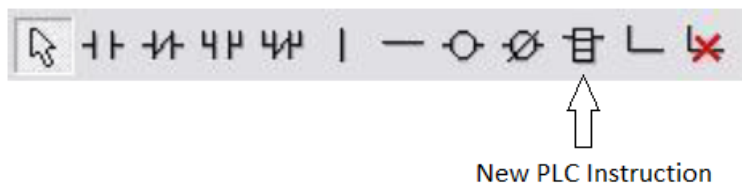


Figura 4.11 – Função *New PLC Instruction*

Algumas das funções especiais mais importantes em programação de PLC's, e as mais utilizadas em aplicações são os temporizadores e os contadores. O funcionamento dessas funções especiais no software CX - Programmer será explicado a seguir.

Temporizadores (TIM)

Na programação do CLP Omron é possível trabalhar apenas com temporizadores na energização (temporizador TON), ou seja, temporizadores com atraso na atuação.

A Figura 4.12 mostra as formas de onda de temporizadores na energização, ilustrando seu princípio de funcionamento. A saída do temporizador TON somente é habilitada se a entrada do temporizador é energizada durante o tempo ajustado. Quando a entrada cai a zero, a contagem é resetada.

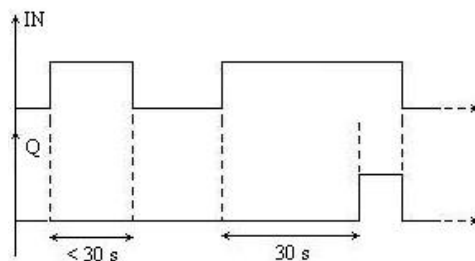


Figura 4.12 – Formas de onda do temporizador TON

A Figura 4.13 mostra a simbologia e operandos dos temporizadores, para configuração do *Preset Value* (PV) nos modos BCD e binário. As instruções TIM ou TIMX(550) operam como temporizadores decrescentes. A configuração da função começa pela definição do número de identificação do temporizado (N), o qual deve assumir valores de 0 a 4095.

O temporizador TIM opera com unidades de 0,1 s. O *range* para o *Set Value* (SV) é de 0 até 999,9. Para isso, os valores que devem ser informados no campo S devem ser ajustados de #0000 a #9999, onde # indica formato BCD.

O temporizador TIMX opera com unidades de 0,001 s (1 ms). Os valores que devem ser informados no campo S devem ser ajustados de #0000 a #FFFF ou de &0 a &65535, onde & indica formato decimal.

Há também o TIMH (*High Speed Timer*) e o TIML(542) (*Long Timer*). O TIMH usa base de tempo de 10 ms, N de 0 a 4095 e S de #0000 a #9999. O TIML(542) usa base de tempo de 0,1 s, N de 0 a 4095 e S de #00000000 a #99999999 ou &00000000 a #4294967294.

PV refresh method	Symbol		Operands
BCD		N: Timer number S: Set value	N: 0000 to 4095 (decimal) S: #0000 to #9999 (BCD)
Binary		N: Timer number S: Set value	N: 00000 to 4095 (decimal) S: &0 to &65535 (decimal) #0000 to #FFFF (hex)

Figura 4.13 – Parâmetros de Configuração e Simbologia para Temporizadores do CLP Omron

Temporizadores de Pulso

Temporizadores de pulso podem ser introduzidos na programação do CLP Omron utilizando funções especiais. Para isso é só inserir contatos convencionais e na sua identificação utilizar uma das funções disponíveis no CLP. Um temporizador de pulso de 1 segundo, por exemplo, pode ser introduzido como no exemplo mostrado na Figura 4.14. Note que, no exemplo mostrado produz-se um pisca-pisca que faz com que a saída seja energizada ou desenergizada a cada 1 segundo.

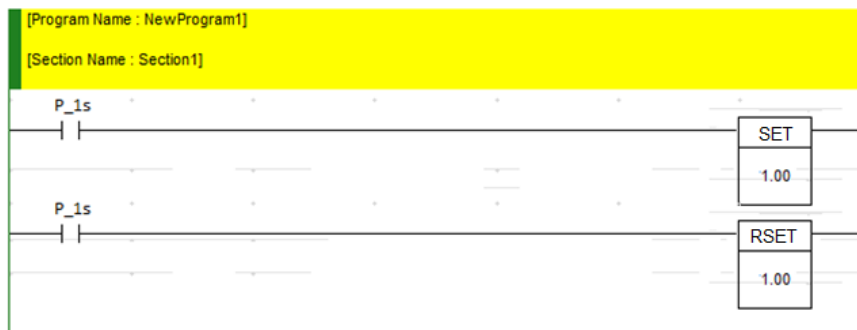


Figura 4.14 – Pisca-pisca utilizando temporizador de pulso

Contadores (CNT)

As funções CNT e CNTX(546) operam como contadores decrementais. O valor de contagem vai de 0 a 9999 para o CNT e de 0 a 65535 para o CNTX(546).

A Figura 4.15 mostra simbologia e os operandos dos contadores para os modos BCD e binário.

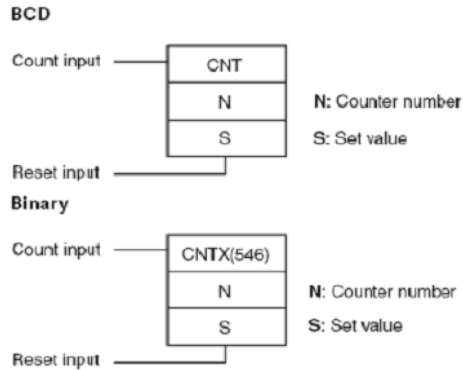


Figura 4.15 – Simbologia para Contadores do CLP Omron

4.4 – Exercícios de Fixação.

Visando fixar os conceitos apresentados serão agora propostos alguns exemplos de aplicação. Sugere-se que o leitor utilize os conceitos apresentados até aqui para desenvolver programas em Ladder que implementem estas aplicações.

Exercício 4.1 – Sistema de Controle de Nível em Um Reservatório

Considere o sistema de reservatório ilustrado na Figura 4.16 composto de uma válvula de entrada P, duas bombas (acionadas por M1 e M2), dois alarmes (AL e AL2) e quatro sensores de nível (a, b, c, d).

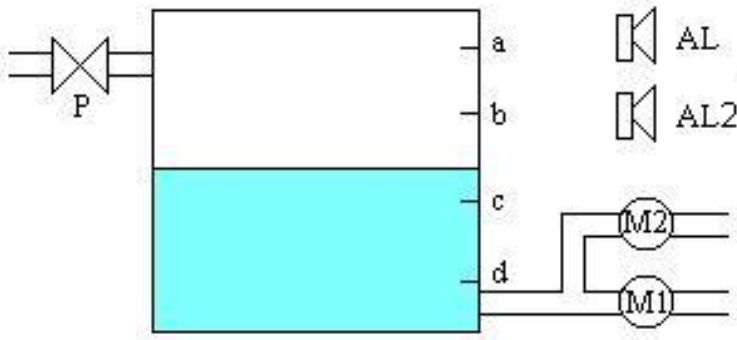


Figura 4.16 – Sistema de Controle de Nível de Um Reservatório

As condições de funcionamento são as seguintes: se o nível for 'a', então fecha-se a válvula P. Se o nível for inferior a 'b', então abre-se a válvula P. Acima de 'b', M1 e M2 bombeiam. Abaixo de 'b', somente M1 bombeia. Abaixo de 'c', soa o alarme AL. Abaixo de 'd', nenhuma das bombas deverá funcionar.

A aplicação também deve analisar possíveis incoerências nas medições dos sensores. Por exemplo, se o nível está acima de 'c', portanto 'c' e 'd' devem estar ligados e 'a' e 'b' desligados. Contudo se 'c' está ligado e 'd' desligado, indica falha no sensor 'd'. Na aplicação deve-se detectar as falhas nos sensores e caso isto ocorra todas as saídas devem ser desligadas (válvula, bombas e alarme AL) e sinalizar utilizando o alarme AL2 para que o operador possa providenciar a manutenção.

Considere a utilização de um CLP Omron para controlar o sistema e a implementação utilizando o software CX-One. Considere também que a conexão com o CLP será efetuada conforme o diagrama de interconexões elétricas apresentado a seguir. Note que no diagrama os sensores de nível (a, b, c e d) são conectados em entradas digitais do CLP (respectivamente I1.0 a I1.3) (Figura 4.17). Observe também que as saídas digitais do CLP são respectivamente:

- O0.0 – Válvula P
- O0.1 – Alarme AL
- O0.2 – Contator que aciona o motor M1
- O0.3 – Contator que aciona o motor M2

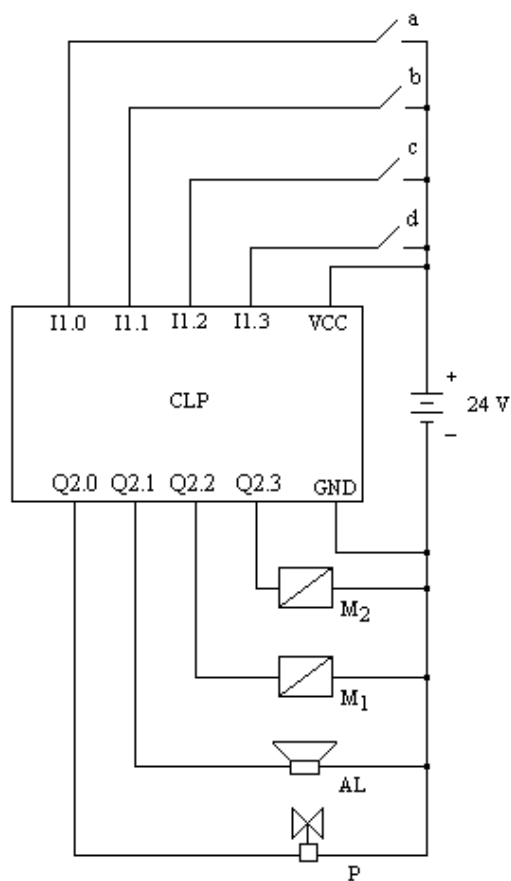


Figura 4.17 – Diagrama de Interconexões Elétricas para Sistema de Controle de Nível

Exercício 4.2 – Partida Estrela-Triângulo de Um Motor de Indução Trifásico

Considere o sistema de partida Estrela-Triângulo de um motor assíncrono trifásico comandado pelos *push-buttons* B₁, B₂ e B₃, ligado conforme o esquema mostrado na Figura 4.18 abaixo:

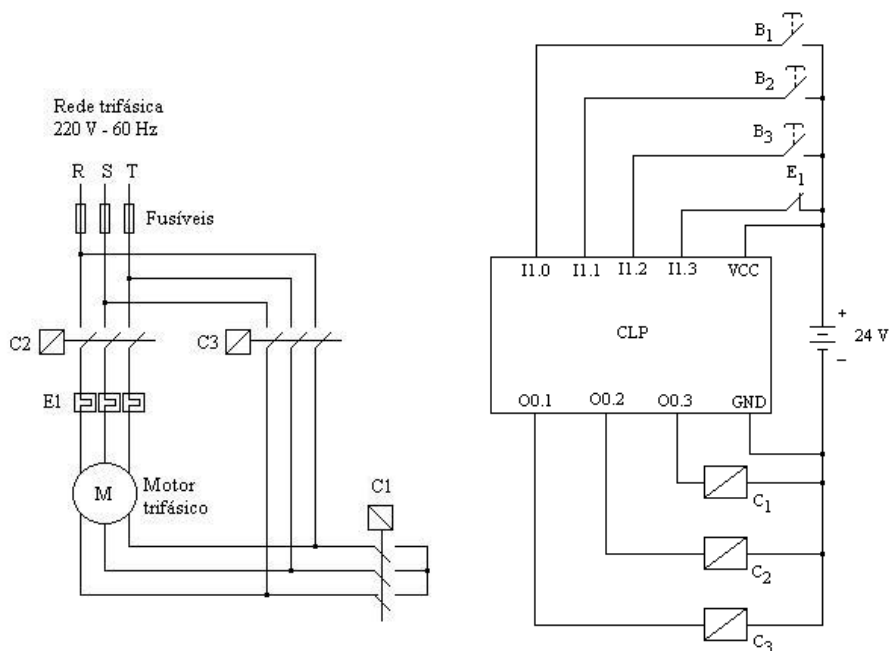


Figura 4.18 – Sistema de Partida Estrela-Triângulo de Um Motor de Indução Trifásico e Diagrama de Interconexões Elétricas

Onde:

B₁ – Botão liga

B₂ – Botão desliga

B₃ – Botão de comutação Y - Δ

E₁ – Relé térmico

C₁ – Contator principal

C₂ – Contator estrela

C₃ – Contator triângulo

O funcionamento do circuito é o seguinte:

- Na partida do motor, comandado pelo botão B₁, os contadores C₁ e C₂ devem ser ativados.
- A comutação é comandada pelo botão B₃. Neste caso C₂ e C₃ devem ser ativados.
- Se houver elevação de temperatura, o relé térmico desabilita o sinal E₁ que pode ser monitorado pelo CLP para desligar o motor.
- O motor pode ser em qualquer momento desligado pelo botão B₂.

O fluxograma da Figura 4.19 pode ser utilizado como auxiliar no projeto.

Cap. 4 – Programação em Ladder no CLP Omron

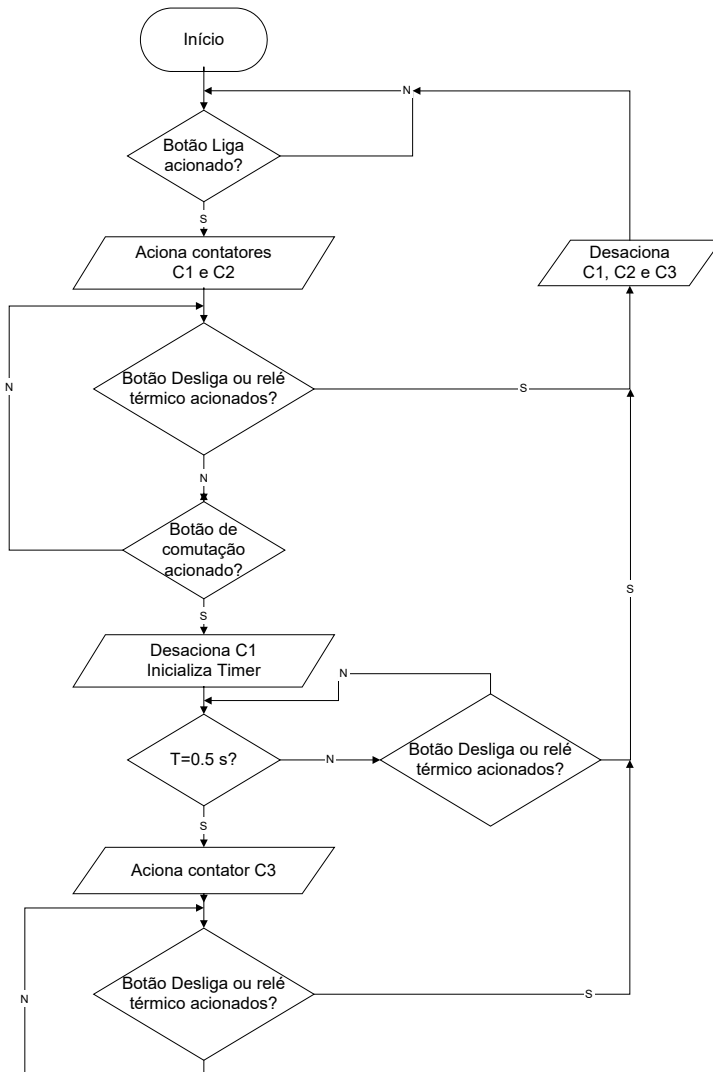


Figura 4.19 – Fluxograma da Aplicação de Partida Estrela-Triângulo de Motor Trifásico

O diagrama das conexões estrela e triângulo e as tensões e correntes nos enrolamentos do motor são apresentados na Figura 4.20. Na figura, U é a tensão entre dois terminais do motor, U_f é a tensão em cada enrolamento, I é a corrente na entrada de cada terminal e I_f é a corrente circulando em cada enrolamento. Note que na partida em estrela, a tensão nos terminais é reduzida para diminuir a corrente circulando no motor. Este efeito irá minimizar o pico de corrente que tipicamente ocorre nos motores, chegando a valores de até 7 vezes o seu valor nominal. Quando o motor entra em regime, utiliza-se a conexão delta, ou triângulo, onde tensão e correntes são mais elevadas.

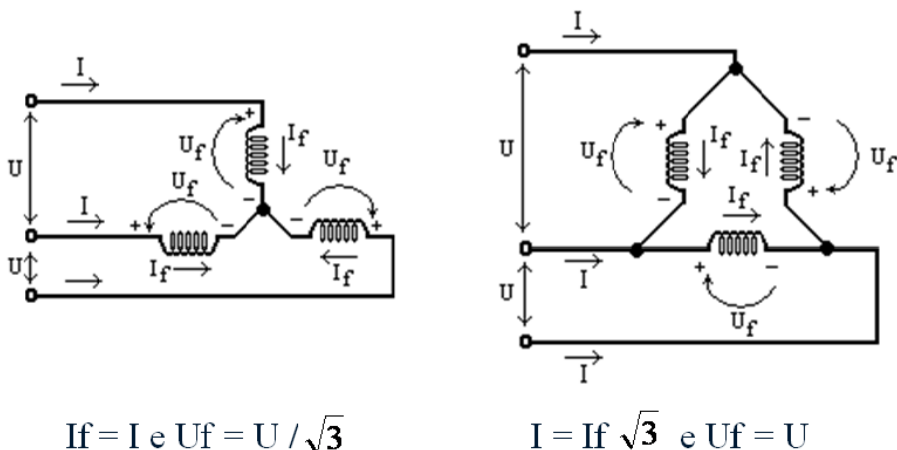


Figura 4.20 – Conexões Estrela e Triângulo no Motor Trifásico

Elabore o programa em ladder utilizando o software CX-One para implementar a partida estrela-triângulo do motor. Lembre-se que os *push-buttons* trabalham com pulsos.

Exercício 4.3 – Furadeira Semiautomática com Contador

Considere a utilização do controlador lógico programável da Omron para controlar o seguinte sistema ilustrado (Figura 4.21).

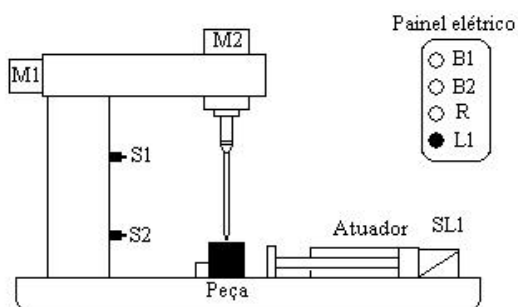


Figura 4.21 – Sistema de Furação de Peças Semiautomático

O processo consiste de uma furadeira semiautomática comandada pelo acionamento de botões. O comando de início é bimanual (botões B1 e B2) sendo efetuado pelo usuário após ter colocado a peça na posição de furação. Se somente um dos dois botões for acionado e o operador não acionar o outro em 3 segundos o processo não poderá ser ligado. Para reconhecer o posicionamento da ferramenta de furação, utiliza-se um sensor que funciona conforme resumido abaixo (Figura 4.22).

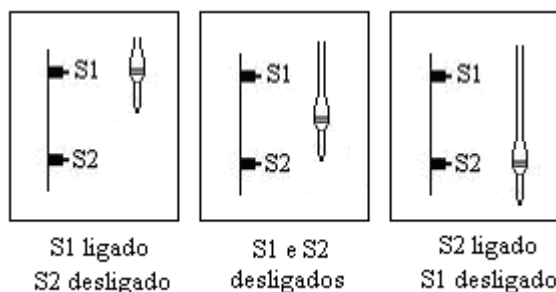


Figura 4.22 – Funcionamento dos sensores na furadeira

O processo funciona somente se a furadeira estiver na posição inicial correta (sensor S1 acionado).

Após o acionamento, a morsa é imediatamente fechada, prendendo a peça. O controle da morsa é efetuado através de um solenoide (SL1). Após 2 segundos o motor M1 deve ser ligado, fazendo a furadeira descer em alta velocidade (comando C1 do motor). Ao ativar o sensor S2 o sistema deve reduzir a velocidade de descida da furadeira (comando C2 do motor) e disparar o processo de furação (motor M2).

Após 3 segundos, o processo de furação é concluído. Neste caso, o motor da broca (M2) deve ser desligado, subindo a furadeira (comando C3 do motor). Ao acionar S1 a morsa deve ser aberta permitindo reinicializar o processo.

Após cada furação efetuada, um contador é incrementado. Após terem sido trabalhadas 4 peças, o processo é paralisado independente da ação do operador sendo isto sinalizado através de um indicador óptico (lâmpada L1).

Para restabelecer o funcionamento normal o usuário deve acionar o botão R para resetar o contador e permitir a continuação do processo de furação.

Entradas:

- Botões B1 e B2 – comando bimanual;
- Sensor S1 – unidade de furação posicionada para início do processo;
- Sensor S2 – posição de furação;
- Botão R – reseta e reinicializa contador.

Saídas:

- Motor M1 – comandos C1(descida em alta velocidade), C2 (descida em baixa velocidade) e C3 (subida);
- Motor M2 – aciona furadeira;
- Solenoide SL1 – aciona morsa;
- Lâmpada L1 – sinaliza contagem atingida.

Elabore o programa em ladder utilizando o software CX-One incluindo temporizadores e contadores para implementar o processo descrito acima. O fluxograma da aplicação está apresentado na Figura 4.23.

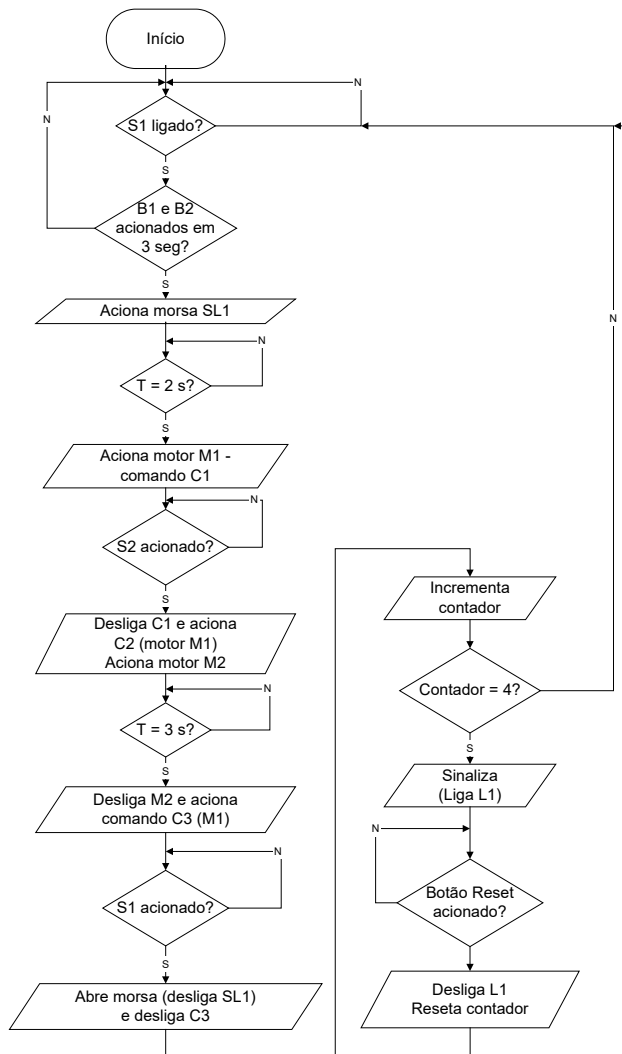


Figura 4.23 – Fluxograma da Aplicação da Furadeira Semiautomática
Exercício 4.4 – Semáforo

Considere a utilização do controlador lógico programável da Omron (software CX-One) para a implementação de um semáforo que funciona da seguinte forma:

- 1 O semáforo deve iniciar a operação após o acionamento do botão B1;
- 2 Após o início de operação o semáforo deve apresentar nos LEDs vermelho (VM), verde (VD) ou amarelo (AM) a seguinte sequência:

VD – 5 segundos

AM – 2 segundos

VM – 8 segundos

Retorna, repetindo a sequência continuamente durante 3 minutos.

Esta sequência representa a operação durante o dia.

- 3 Após 3 minutos, o sistema deverá apresentar a operação com apenas o LED amarelo sendo acionado numa frequência de 1 Hz. Esta situação representa a operação dos semáforos à noite e ocorrerá automaticamente, sendo mantida durante 2 minutos.
- 4 Deseja-se que a situação descrita no item 3 possa também ser comandada pelo operador através do acionamento de um botão, por exemplo, quando houver algum problema no tráfego ou para manutenção. Neste caso, a autoridade de tráfego deve acionar o botão B2 em qualquer momento do dia. Para restabelecer a operação normal o operador deve novamente acionar o botão B2. É importante lembrar que, na ocorrência da situação descrita anteriormente, o contador de

tempo que mantém a operação conforme o item 2 durante 3 minutos não deve ser afetado.

- 5 Após 2 minutos, o sistema continua a operação ciclicamente, repetindo a sequência descrita no item 2.

OBS – Os intervalos de tempo propostos no projeto, nem de longe equivalem aos tempos utilizados comumente na prática. O objetivo é simplesmente permitir que, em laboratório, a implementação do projeto possa ser avaliada mais facilmente.

Exercício 4.5 – Reversão do Sentido de Circulação de um Veículo Acionado por Motor de Indução

Deseja-se automatizar um sistema constituído por um veículo que se movimenta através de uma linha férrea sendo comandado pelo acionamento de um motor de indução conforme ilustrado na Figura 4.24 abaixo:

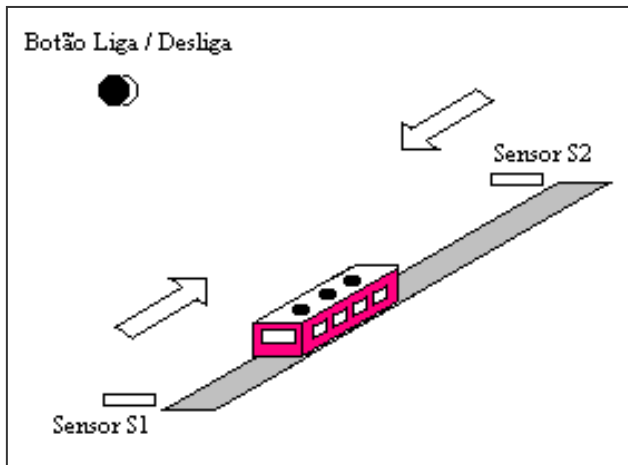


Figura 4.24 – Sistema de Controle de um Veículo em Linha Férrea

O sistema é inicializado através do acionamento de um *push-button* (botão Liga/Desliga) a partir do qual é feito o acionamento do motor do veículo, fazendo com que este se movimente para a extremidade direita da linha.

O veículo é paralisado automaticamente a partir da detecção do acionamento do sensor S_2 permanecendo assim durante 20 segundos.

Após este tempo o veículo é acionado automaticamente, movimentando-se para a extremidade esquerda da linha.

O veículo é paralisado novamente a partir da detecção do acionamento do sensor S_1 permanecendo assim durante outros 20 segundos.

Após este tempo o veículo é acionado novamente, movimentando-se outra vez para a extremidade esquerda da linha.

O funcionamento é intermitente, sendo somente interrompido se houver um novo acionamento no botão Liga/Desliga. Observe que em se tratando de um *push-button*, a operação Liga ou Desliga é invertida a cada acionamento neste botão.

Considere que o desligamento pode ocorrer em qualquer instante, podendo ocorrer mesmo na região intermediária, mas em qualquer caso, o veículo irá paralisar aguardando um novo comando.

Entradas:

- Botão Liga-Desliga – botão pulsador;
- Sensores S_1 e S_2 .

Saídas:

- Motor M1 controlado pelos contadores C1 (movimentação da esquerda para direita) e C2 (movimentação da direita para esquerda). O diagrama esquemático do circuito de acionamento do motor é ilustrado na Figura 4.29.

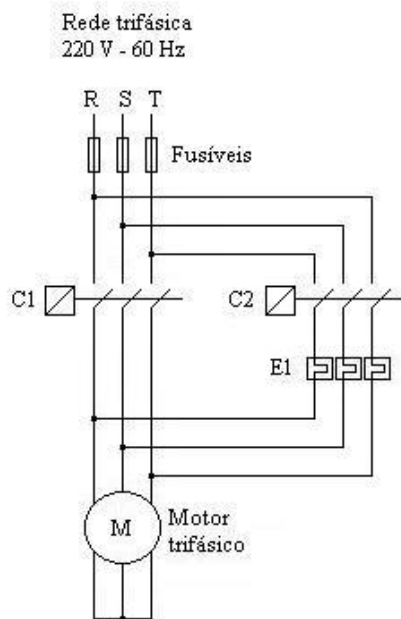


Figura 4.25 – Diagrama de Interconexões Elétricas para o Sistema de Reversão do Sentido de Circulação de um Motor de Indução

Elabore o programa em ladder utilizando o software CX-One para implementar o processo descrito acima.

Exercício 4.6 – Semáforos em Cruzamento

Elaborar um programa para implementação utilizando o CLP Omron (software CX-One) para controlar um sistema constituído de dois semáforos num cruzamento conforme detalhado abaixo e ilustrado na Figura 4.30. Dependendo da estratégia utilizada, podem ser utilizados

registradores, comparadores, operações aritméticas ou outras funções especiais para simplificar a programação.

- 1 Os semáforos devem iniciar a operação após o acionamento do botão B1.
- 2 Os semáforos apresentam funcionamento diferente dependendo do período: durante o dia ou à noite.

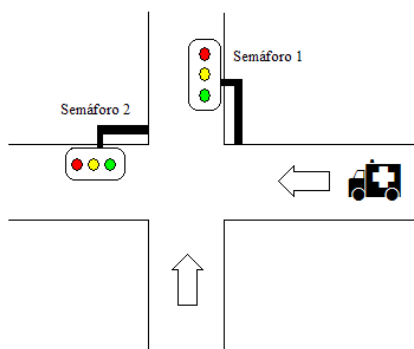


Figura 4.26 – Uma aplicação utilizando semáforos num cruzamento

- 3 A condição inicial para os semáforos após o acionamento do botão é equivalente à operação do dia devendo apresentar nos LED's vermelho (VM1 e VM2), verde (VD1 e VD2) ou amarelo (AM1 e AM2) a seguinte sequência:

Sequência: Semáforo 1

Semáforo 2

1 VD1 – 5 segundos VM2 – 7 segundos

2 AM1 – 2 segundos

3 VM1 – 8 segundos VD2 – 6 segundos

4 AM2 – 2 segundos

Retorna, repetindo a sequência continuamente durante 3 minutos.

- 4 Após 3 minutos, o sistema deverá apresentar a operação com apenas os LEDs amarelos sendo acionados numa frequência de 1 Hz. Esta situação representa a operação dos semáforos à noite e ocorrerá automaticamente, sendo mantida durante 2 minutos.
- 5 Deseja-se que a situação descrita no item 4 possa também ser comandada pelo operador através do acionamento de um botão, por exemplo, quando houver algum problema no tráfego ou para manutenção. Neste caso, a autoridade de tráfego deve acionar o botão B2 em qualquer momento do dia. Para restabelecer a operação normal o operador deve novamente acionar o botão B2. É importante lembrar que, na ocorrência da situação descrita anteriormente, o contador de tempo que mantém a operação conforme o item 3 durante 3 minutos não deve ser afetado.
- 6 Após 2 minutos, o sistema continua a operação ciclicamente, repetindo a sequência descrita no item 3.

OBS – Os intervalos de tempo propostos no projeto, nem de longe equivalem aos tempos utilizados comumente na prática. O objetivo é simplesmente permitir que, em laboratório, a implementação do projeto possa ser avaliada pelo professor.

5 – PNEUMÁTICA

Neste capítulo serão abordados os fundamentos da pneumática (ATTIÉ, 1998). Na pneumática podem ser desenvolvidas uma infinidade de aplicações onde utiliza-se o controle através de CLPs. Daí a importância de se conhecer os elementos básicos da pneumática tais como cilindros atuadores, válvulas direcionais e sensores, essenciais para a implementação dos mais simples circuitos pneumáticos.

5.1 – Introdução à Pneumática

Consiste na área da mecânica em que o ar comprimido é utilizado como meio de comando e trabalho para, baseado nas leis da física, produzir movimento e força (energia mecânica).

A palavra pneumática vem do grego “pneuma” que significa respiração, vento, e trata do comportamento dos gases.

O ar comprimido foi introduzido no meio industrial após 1950 sendo utilizado na automação e na racionalização da força humana para trabalhos cíclicos e metódicos. Antes porém já existiam alguns campos de aplicação da pneumática, como por exemplo, a indústria de mineração, a construção civil e a indústria ferroviária (freios a ar comprimido).

5.2 – Propriedades do ar comprimido.

A utilização do ar comprimido na indústria apresenta as seguintes vantagens:

Cap. 5 – Pneumática

- Quantidade: o ar comprimido encontra-se em quantidades ilimitadas praticamente em todos os lugares.
- Transporte: Facilmente transportável por tubulações, mesmo para distâncias consideravelmente grandes, sem necessidade de preocupação com o retorno do ar.
- Armazenamento: O ar pode ser sempre armazenado em um reservatório.
- Temperatura: O trabalho realizado com ar comprimido é insensível às oscilações da temperatura. Isto garante um funcionamento seguro mesmo em situações térmicas extremas.
- Segurança – Se o ar comprimido é mantido livre de impurezas tais como óleo, não existe o perigo de explosão.
- Limpeza – O ar comprimido é limpo. O ar que eventualmente escapa das tubulações ou outros elementos inadequadamente vedados, não polui o ambiente. Estas características tornam o seu uso bastante difundido em indústrias alimentícias, madeiras, têxteis e químicas.
- Construção dos elementos – Os elementos de trabalho são de construção simples e, portanto, relativamente baratos.
- Velocidade – O ar comprimido é um meio de trabalho rápido, permitindo alcançar altas velocidades de trabalho.
- Regulagem – As velocidade e forças de trabalho dos elementos a ar comprimido são reguláveis.

As principais limitações no uso do ar comprimido são:

- Preparação – O ar comprimido requer uma boa preparação. Impurezas e umidade devem ser evitados, pois provocam desgastes nos elementos pneumáticos.
- Compressibilidade – Não é possível manter uniforme e constante a velocidade dos cilindros e motores mediante ar comprimido.
- Forças – O ar comprimido é econômico somente a uma determinada força, limitada pela pressão normal de trabalho 700 KPa (7 bar), e também pelo curso da velocidade. O limite está fixado entre 20.000 e 30.000 N.
- Escape de ar – O escape de ar é ruidoso. Com o desenvolvimento de silenciadores, este problema está atualmente resolvido.

5.3 – Sistema de Produção e Distribuição de Ar Comprimido.

O sistema de produção e distribuição de ar comprimido consiste de uma série de elementos utilizados para produzir, armazenar e preparar o ar comprimido, garantindo o fornecimento com pressão e grau de pureza adequados para o consumo em aplicações. É importante destacar que um grande número de dispositivos torna-se necessário para eliminar umidade/impurezas que podem danificar ou provocar corrosão nos elementos pneumáticos, tubulações, etc., aumentando a vida útil destes componentes, mas também por segurança (por exemplo, a mistura gasosa de ar e óleo pode provocar explosões quando atinge acima de 80° C). O sistema envolve os seguintes dispositivos:

5.3.1 – Compressores

São os elementos que absorvem o ar atmosférico comprimindo-o para a pressão de trabalho desejada. Podem ser de vários tipos: compressores de êmbolo de movimento linear, compressores rotativos e turbo-compressores.

Compressor de êmbolo ou pistão (compressores de êmbolo de movimento linear) baseia-se no princípio da redução de volume. Consegue-se a compressão sugando o ar para um ambiente fechado e diminuindo posteriormente o tamanho deste ambiente. É o mais utilizado, sendo apropriado para baixas e médias pressões, podendo também ser utilizado para altas.

Alguns compressores de êmbolo apresentam dois ou mais estágios, onde o ar aspirado é comprimido pelo primeiro pistão, refrigerado imediatamente, para logo ser comprimido pelo segundo pistão. Além disso permite pressões mais elevadas:

Até 400 KPa	1 estágio
Até 1500 KPa	2 estágios
Acima de 1500 KPa	3 estágios

Alguns compressores de êmbolo apresentam membrana. Neste compressor, uma membrana separa o êmbolo da câmara de trabalho fazendo com que o ar não tenha contato com as peças móveis, estando assim isento de óleo. Estes compressores são utilizados em indústrias alimentícias, farmacêuticas e químicas.

Compressores rotativos – Seu funcionamento baseia-se no acionamento de eixos giratórios. São exemplos: o compressor radial de palhetas, o compressor de parafusos helicoidais e o compressor Roots.

- **Turbo-compressores** – Funcionam segundo o princípio de fluxo, ou seja, sucção de ar de um lado e compressão no outro por aceleração de massa (turbina).

5.3.2 – Unidade de Preparação, Resfriamento e Distribuição de Ar Comprimido

Unidade de resfriamento – O processo nos compressores produz aumento da temperatura do ar. A unidade de resfriamento é responsável pela redução da temperatura do ar comprimido através de trocadores de calor.

Separador de água e óleo – Elimina a umidade, impurezas (como a poeira) e resíduos de óleo no ar comprimido produzido pelo compressor.

Reservatório de armazenamento – Armazena o ar, minimiza oscilação na pressão e condensa a umidade contida no ar em razão do resfriamento permitindo retirá-la.

Secadores – Se necessário são introduzidos secadores de ar comprimido para minimizar a umidade presente no ar comprimido, podendo a secagem ser efetuada através de um processo químico (secagem por absorção), processo regenerativo (por adsorção usando sílica gel) ou por resfriamento (secagem a frio – máquina de refrigeração).

Drenos – Consiste de válvulas automáticas para eliminar a umidade na rede de distribuição.

Unidade de conservação – Unidade próxima do ponto de consumo e engloba os seguintes dispositivos:

a) Filtro para ar comprimido – Apresentam um elemento filtrante (poroso) com baixa porosidade (micro poros) para retenção de partículas de impurezas e umidade.

b) Regulador de pressão e manômetro – Para regular e medir a pressão do ar comprimido, mantendo pressão de trabalho constante. Segundo estudos, a pressão mais adequada para trabalhar é de 6 a 7 bar.

c) Copo condensador de água – Acumula a água proveniente do ar comprimido.

d) Lubrificador de ar comprimido – Abastece suficientemente com material lubrificante os elementos pneumáticos. O elemento lubrificante é necessário para garantir um menor desgaste dos elementos móveis minimizando as forças de atrito e protegendo os aparelhos contra corrosão. Os lubrificadores de ar trabalham com o princípio de Venturi em que no ponto de estrangulamento há uma queda de pressão permitindo sugar óleo de um reservatório e misturá-lo com o ar em forma de neblina.

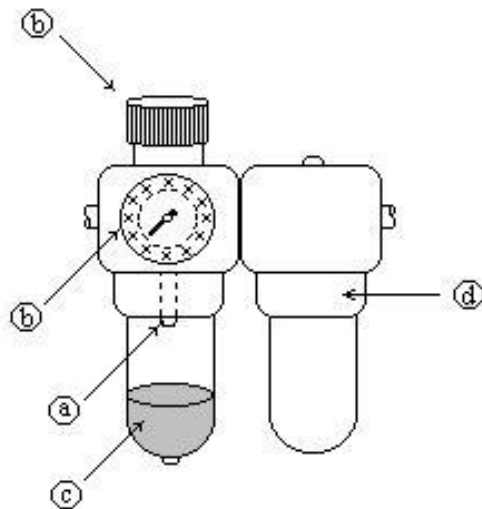


Figura 5.1 – Unidade de Conservação

5.4 – Estrutura de Sistemas de Controle Pneumáticos.

Sistemas de controle pneumáticos envolvem uma série de dispositivos dependendo da natureza e da complexidade do projeto. Um circuito pneumático completo apresenta o diagrama de blocos conforme ilustrado a seguir.

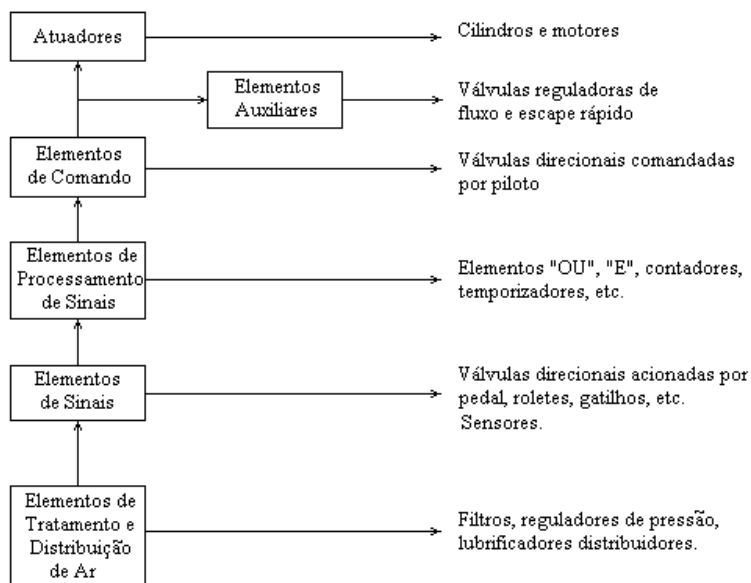


Figura 5.2 – Diagrama de Blocos de Sistemas de Controle Pneumáticos

Em sistemas mais simplificados, similares aos que serão vistos nas experiências em laboratório, apenas alguns elementos tais como atuadores, válvulas direcionais e elementos de tratamento e distribuição de ar estão presentes.

No Anexo I é apresentada a simbologia de alguns dispositivos utilizados em sistemas pneumáticos e Eletropneumáticos.

5.5 – Atuadores.

Atuadores ou elementos pneumáticos de trabalho, são elementos responsáveis pela transformação da energia pneumática (ar comprimido sob pressão) em movimento e força (energia mecânica). Os atuadores podem executar movimentos lineares ou rotativos. São exemplos de atuadores lineares os cilindros de simples ação, cilindros de dupla-ação e alguns cilindros especiais, tais como: cilindro de dupla ação com amortecimento regulável de fim de curso, cilindro com haste passante, cilindros de múltiplas posições, etc. São exemplos de atuadores rotativos os motores de giro contínuo e cilindros de giro limitado.

O cilindro é o elemento pneumático mais conhecido e difundido por sua construção simples e robusta e por permitir amplas possibilidades de utilização. Os campos de aplicação mais comuns e importantes dos cilindros pneumáticos são fixação, transporte e usinagem de peças.

Alguns exemplos de construção de cilindros:

Cilindros de simples ação.

Em cilindros de simples ação a atuação ocorre pela pressão do ar comprimido, sendo o retorno pela força da mola ou por força externa quando o ar comprimido é retirado.

O curso dos cilindros de simples ação está limitado ao comprimento da mola. Por esta razão não são fabricados estes tipos de cilindros com mais de 100 mm.

São especialmente utilizados em operações que envolvam fixação, expulsão, extração e prensagem, entre outros.

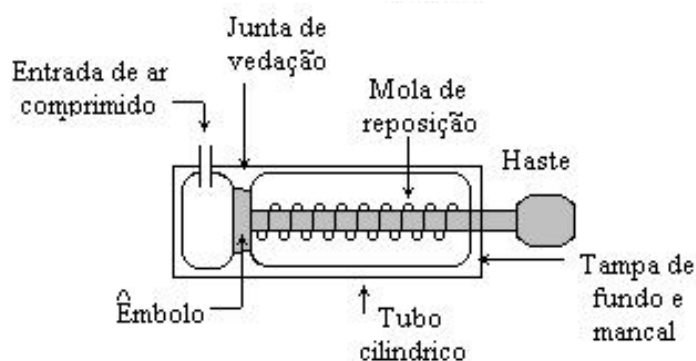


Figura 5.3 – Cilindro de Simples Ação

Cilindros de dupla-ação.

Em cilindros de dupla ação, as forças retílineas em duas direções são executadas por ar comprimido, sem necessidade de mola de reposição ou força externa.

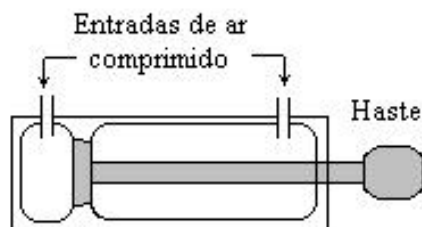


Figura 5.4 – Cilindro de Dupla-Ação

5.6 – Válvulas Direcionais.

São os componentes do sistema que direcionam o ar comprimido pelas vias, desviando-o para onde for mais conveniente em um determinado momento por ação de um acionamento externo.

O acionamento pode ser:

- Manual (muscular) – Por botão, alavanca ou pedal.
- Mecânico – Por mola, rolete apalpador, came, rolete escamoteável, etc.
- Elétrico – Por eletroímã com enrolamento ativo (bobina solenoide), por motor elétrico, etc.
- Pressão – Por aumento de pressão ou por alívio de pressão, por pressão diferencial, etc.

A representação dos vários tipos de válvulas disponíveis depende do número de vias, número de posições de trabalho e do tipo de acionamento.

As vias (conexões) das válvulas dividem-se em:

- Conexão de entrada de ar comprimido (pressão);
- Conexão para alimentação dos atuadores pneumáticos (utilização);
- Orifícios de escape.

As vias são identificadas através de letras maiúsculas (norma DIN ISSO 1219) ou números (norma DIN ISSO 5599).

Exemplos:

Válvula 3/2 vias normal fechada (NF), acionamento por botão, retorno por mola.

Usada para comandar o acionamento de cilindros de ação simples.

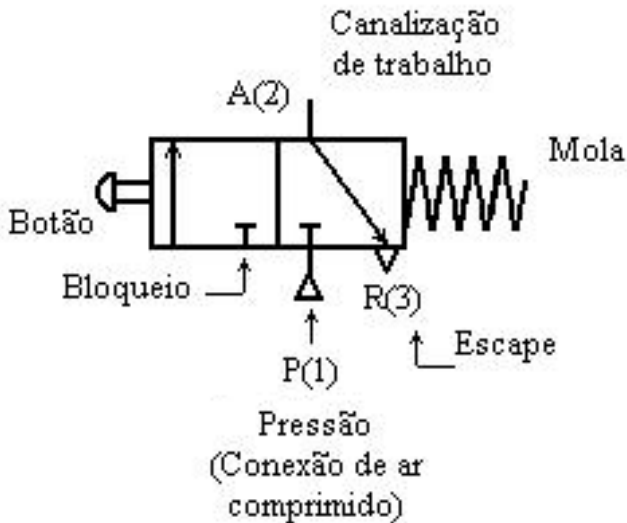


Figura 5.5 – Representação da Válvula 3/2 Vias NF, Acionamento por Botão, Retorno por Mola

Válvula 3/2 vias normal aberta (NA), acionamento por solenoide, retorno por mola.

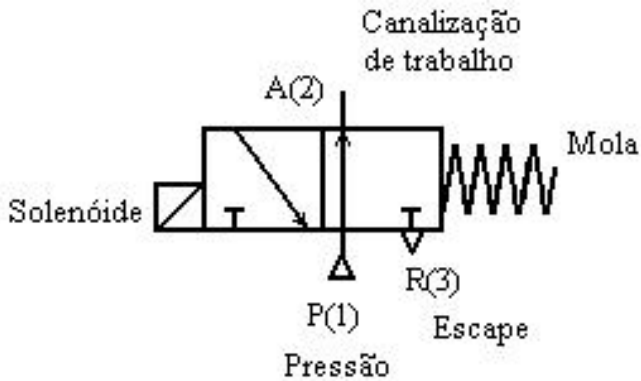


Figura 5.6 – Representação da Válvula 3/2 Vias NF, Acionamento por Solenoide, Retorno por Mola

Válvula 5/2 vias, acionamento por duplo solenoide.



Figura 5.7 – Representação da Válvula 5/2 Vias, Acionamento por Duplo Solenoide

Outras válvulas:

- Válvulas de bloqueio (Ex.: válvulas de simultaneidade, válvulas alternadoras, válvulas de escape rápido, etc.)
- Válvulas reguladoras de pressão.
- Válvulas reguladoras de fluxo.
- Válvulas de rolete.

5.7 – Sensores.

São elementos de sinais utilizados para detectar proximidade ou passagem de atuadores ou peças móveis em uma série de aplicações na mecânica sendo o funcionamento baseado em características elétricas específicas.

- Sensores capacitivos – Utiliza como princípio de funcionamento a variação do dielétrico. Um oscilador alimenta um capacitor formado por duas placas em sua extremidade, que é a parte sensível do componente. Quando algum material ingressa nesta região, provoca uma variação da capacitância atuando sobre o circuito de acionamento. Podem ser usados para detecção de qualquer tipo de material, tais como: papel, madeira, plásticos, farinha, metais e etc.

Sensores indutivos – Utiliza um oscilador de rádio frequência, cuja oscilação é modificada quando algum corpo metálico corta o campo magnético da bobina. Não é necessário o contato do corpo com o sensor. Esta modificação na oscilação é interpretada em um circuito comparador, que irá ativar o gate de um tiristor, tal como uma chave liga/desliga em estado sólido.

Sensores ópticos – Incorpora no mesmo componente um LED e um fototransistor sendo que, quando há proximidade de algum corpo, a luz emitida pelo LED é refletida na superfície deste corpo sendo captada pelo fototransistor.

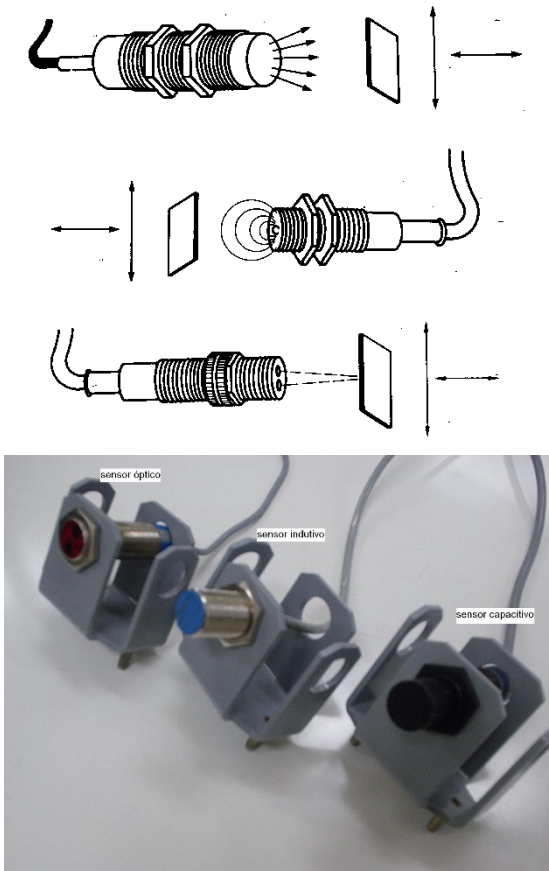


Figura 5.8 – Sensores capacitivo, indutivo e óptico

5.8 – Exemplos de Circuitos Pneumáticos e Eletropneumáticos.

Para ilustrar e para melhor entendimento serão apresentados alguns exemplos de circuitos pneumáticos. O objetivo é compreender o funcionamento destes circuitos e dos elementos utilizados na implementação.

a)

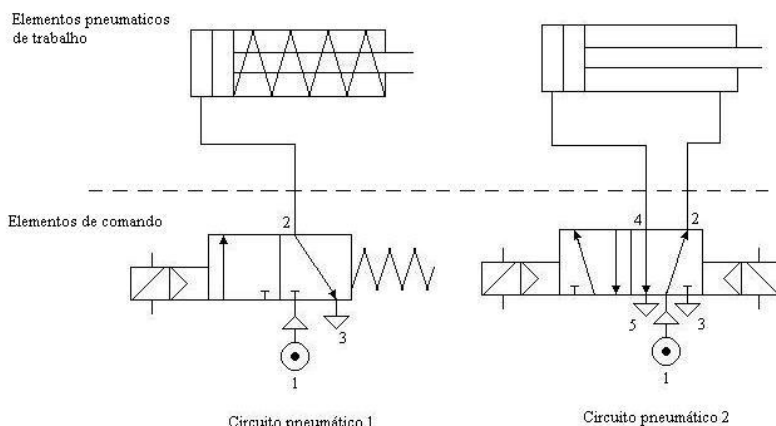


Figura 5.9 – Circuitos pneumáticos básicos.

O circuito pneumático 1 da Figura 5.9 apresenta o acionamento de um cilindro de simples ação comandado através de uma válvula direcional 3/2 vias normal fechada, acionamento por solenoide ou piloto, retorno por mola.

Este circuito poderia ser controlado através da saída de um CLP comandando a tensão aplicada na bobina do solenoide ou através da aplicação de ar comprimido no comando piloto da válvula. Quando isto acontecer ocorre uma comutação na válvula direcional (que equivale ao deslocamento do símbolo da válvula para a direita). Isto faz com que seja aberto o caminho para passagem de ar comprimido que é então direcionado para o cilindro de simples ação. Neste cilindro o ar comprimido pressiona o êmbolo contra a mola efetuando então o acionamento.

Para obter o recuo do cilindro, basta tirar a tensão do solenoide, pois neste caso a válvula é comutada automaticamente para a posição inicial (posição de repouso) devido a ação da mola. Como consequência, devido à ausência de pressão no cilindro, este recua também devido à ação de sua mola.

O circuito pneumático 2 da Figura 5.9 apresenta o acionamento de um cilindro de dupla ação comandado através de uma válvula direcional 5/2 vias, acionamento por duplo solenoide ou piloto.

Acionando o solenoide da esquerda ocorre a comutação da válvula que direciona o ar comprimido e aciona o cilindro. Um simples pulso é suficiente, pois mesmo sem tensão aplicada, a válvula permanece comutada. Acionando o solenoide da direita, ocorre uma nova comutação na válvula que redireciona o ar comprimido fazendo com que o cilindro recue. Portanto este circuito poderia ser controlado simplesmente através do controle de duas saídas de um CLP.

Como demonstrado na Figura 5.2, a estrutura de sistemas de controle pneumáticos envolve uma série de dispositivos dependendo da natureza e da complexidade do projeto. O circuito ilustrado na Figura 5.9 consiste de um circuito simplificado que apresenta elementos pneumáticos de trabalho e elementos de comando utilizados para executar funções específicas.

b)

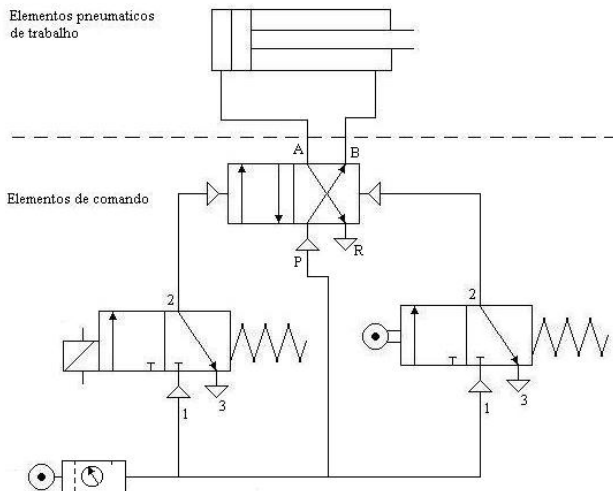


Figura 5.10 – Uma implementação utilizando circuito pneumático

Os elementos que constituem o circuito são:

- Fonte de pressão e manômetro;
- Válvula direcional 3/2 vias, NF, acionamento por solenoide, retorno por mola (Válvula 1);
- Válvula direcional 3/2 vias, NF, acionamento por rolete, retorno por mola (Válvula 2);
- Válvula direcional 4/2 vias, acionamento por piloto (Válvula 3);
- Cilindro de dupla ação.

O princípio de funcionamento pode ser assim descrito:

- Na posição de repouso o ar comprimido é direcionado de forma que o cilindro de dupla ação se encontra recuado.
- Aplicando tensão no solenoide da Válvula 1 esta é comutada fazendo com que o ar comprimido seja aplicado no piloto da Válvula 3. Esta válvula é então comutada permitindo a passagem de ar comprimido que aciona o cilindro atuador.
- O cilindro permanece atuado até que ocorra a comutação da Válvula 2. Isto somente ocorrerá se houver algum acionamento no contato tipo rolete (acionamento mecânico). Caso isto ocorra, o ar comprimido é direcionado para o piloto da Válvula 3, que é comutada, voltando à posição de repouso. Como consequência, o cilindro de dupla ação é recuado devido ao redirecionamento do fluxo de ar comprimido.

6 – PROGRAMAÇÃO DO CLP CPX-CEC - FESTO®

Neste capítulo serão apresentados os aspectos de programação do CLP Festo CPX-CEC o qual será utilizado para desenvolver aplicações de pneumática e hidráulica. Daremos ênfase à programação em Ladder do CLP utilizando o Software CodeSys. Algumas aplicações típicas na área de pneumática serão analisadas e implementadas.

6.1 – Hardware do Controlador Lógico Programável CPX-CEC.

O CLP CPX-CEC (Figura 6.1) é um controlador didático da Festo, sendo que a unidade disponível em laboratório permite controlar dois módulos de entrada digital e dois módulos de saída digital. Pode ser programado diretamente (*stand alone*) ou remotamente usando rede Ethernet ou Fieldbus. Permite a programação utilizando funções desenvolvidas em STL (*Statement List*), ST(*Structured Text*), SFC (*Sequential Flow Chart*), FD (*Function Diagram*) e LDR (*Ladder*). Possui CPU com processador de 400 MHz, 32 MB de RAM e 32 MB de memória Flash. O CLP possui memória de programa de 4 MB e permite tempo de processamento de cerca de 200µs/1k instruções. Utiliza fonte de alimentação de 24 V com corrente de consumo típica de saída de 85 mA para tensão nominal.



Figura 6.1 – PLC Festo CPX-CEC

6.2 – Programação do Controlador Lógico Programável Festo CPX-CEC.

Nesta seção são apresentadas instruções para a programação do CLP Festo utilizando o software CoDeSys Versão 2.3.

CoDeSys é um acrônimo para *Controller Development System* e trata-se de um ambiente de programação de controladores lógicos programáveis seguindo o padrão IEC 61131-3 (Internacional Industrial Standard). Todas cinco linguagens de programação definidas na norma podem ser programadas usando o CoDeSys, a saber:

- IL (*Instruction List*), uma linguagem de programação textual;
- ST (*Structured Text*), também linguagem de programação textual, similar à programação em Pascal ou C;
- LD (*Ladder Diagram*), com programação gráfica baseada na utilização de contatos e bobinas;
- FBD (*Function Block Diagram*), linguagem gráfica que facilita muito o desenvolvimento de sistemas com lógica Booleana e expressões análogas;
- SFC (*Sequential Function Chart* ou *Sequential Flow Chart*), linguagem gráfica conveniente para desenvolvimento de aplicações para sistemas sequenciais.

Adicionalmente o CoDeSys permite a programação utilizando outra linguagem não definida na norma IEC:

- CFC (*Continuous Function Chart*), é uma espécie de editor de FBD orientado para a rede, onde as conexões entre entradas, operadores e saídas são definidas automaticamente, sendo “desenhadas” pelo programador. Todas as caixas podem ser colocadas livremente, o que possibilita programar loops de *feedback* sem variáveis intermediárias.

O CoDeSys não é uma linguagem proprietária da FESTO, o que faz com que o ambiente de desenvolvimento pode ser empregado na programação de diversos outros modelos de PLC. As mais importantes famílias de CPU de 16 e 32 bits podem ser programadas pelo CoDeSys,

tais como C166, TriCore, 80x86, ARM/Cortex, Power Architecture, SH, MIPS, BlackFin, entre outros. Assim, o integrador que faz parte do pacote compila o código e transforma em código binário, o qual pode ser transferido para essas CPUs, desde que, utilizando os drivers específicos.

O sistema inclui ainda a possibilidade de programar diversos tipos de redes de comunicação. Para isso a ferramenta integra protocolos que permitem configurar redes como PROFIBUS, CANOpen, EtherCAT, PROFINET e EtherNet / IP.

Por fim, mais de 250 fabricantes de dispositivos de diferentes setores industriais oferecem dispositivos de automação inteligentes com uma interface de programação CoDeSys. São mais de 400 dispositivos programáveis de fabricantes líderes da área de engenharia de controles, automação e sistemas embarcados. São exemplos: Festo, Schneider Automation, KEB Automation KG, EATON Automation AG, Bosch Rexroth, NEXCOM, Hitachi, Advantech Europe, Mitsubishi Electric Europe, Altus Sistemas de Informática, entre outros.

Tutorial – Desenvolvendo uma Aplicação em Ladder

A programação do CLP CPX-CEC da Festo é efetuada diretamente através do software CoDeSys (Figura 6.2) instalado num PC (*Personal Computer*). Para entrar no software Codesys 2.3 clique em Iniciar / Todos os Programas/ Festo Software / CoDeSys V2.3 by Festo. A tela inicial está apresentada na Figura 6.3.

Para criar um novo programa clique em *File / New* e em *Configuration* selecione o modelo do CLP Festo CPX-CEC / CPX-CEC-C1 (Figura 6.4).

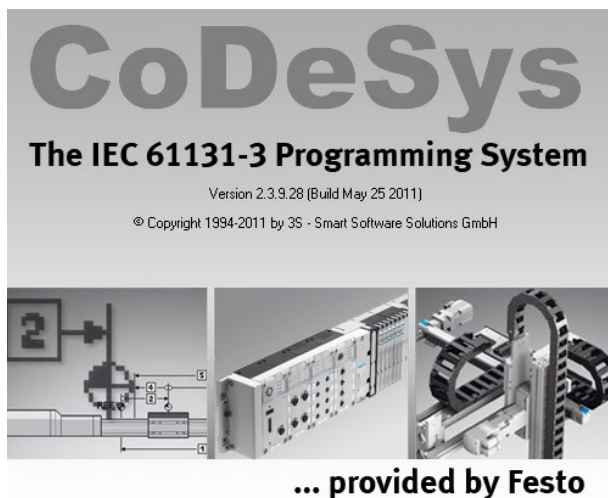


Figura 6.2 – Software CoDeSys Versão 2.3

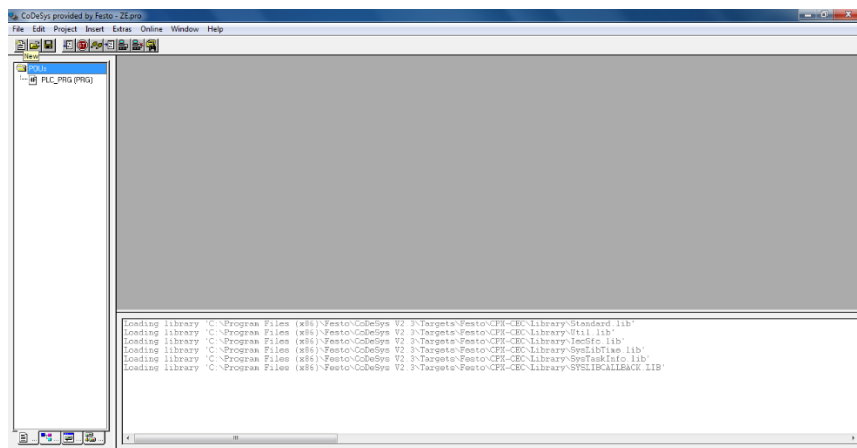


Figura 6.3 – Tela Inicial do Software CoDeSys Versão 2.3

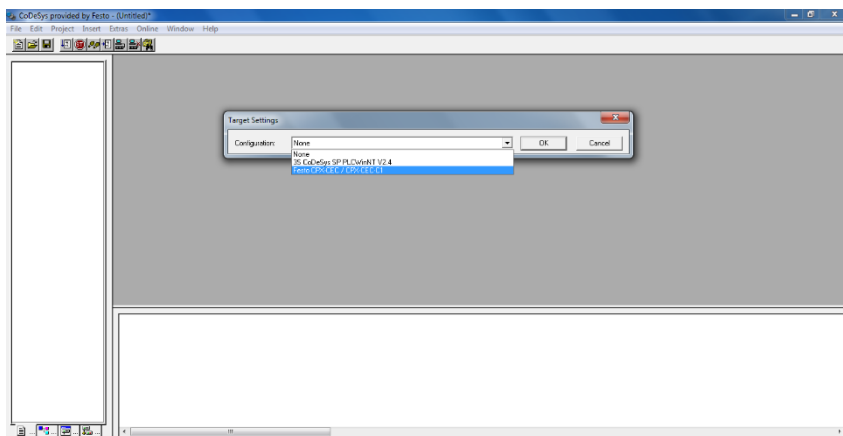


Figura 6.4 – Seleção do Modelo do CLP

Em seguida selecione a linguagem de programação (Figura 6.5), tais como Ladder (LD), Instruction List (IL), Function Blocks Diagram (FBD), SFC (Sequential Flow Chart), entre outras.

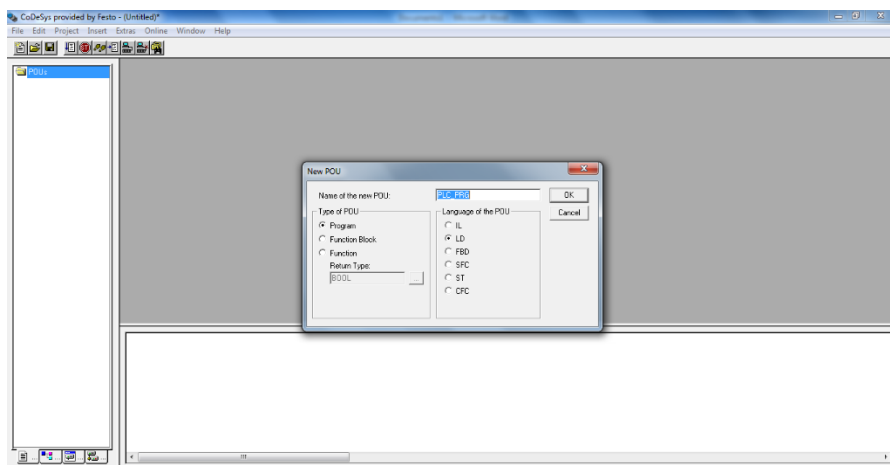


Figura 6.5 – Seleção da Linguagem de Programação

A tela de programação em Ladder é apresentada na Figura 6.6. Contudo, antes de começar a programar precisamos configurar o CLP e seus módulos. A conexão com o CLP é feita pela rede Ethernet utilizando o IP do computador. Para isso, precisamos primeiramente localizar o atalho para realizar as configurações em “Resources” (na lateral esquerda e inferior da tela), como mostrado a seguir.

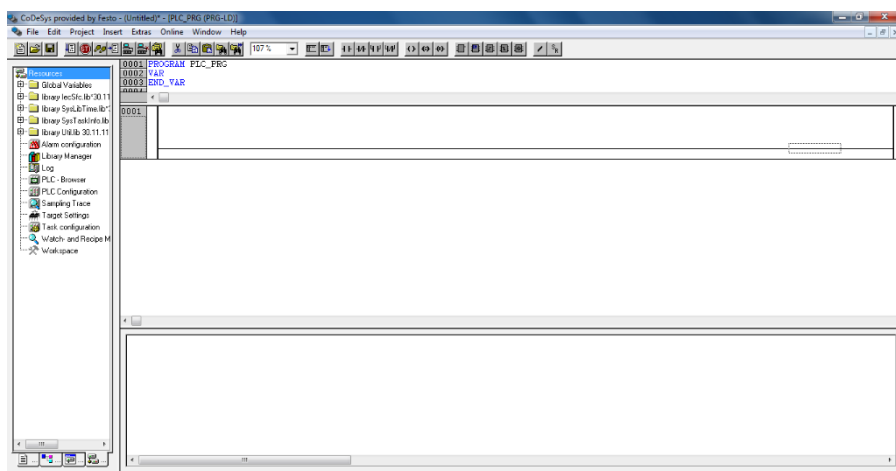


Figura 6.6 – Tela de Programação em Ladder

Dentre as opções disponíveis em “Resources”, temos “PLC Configuration” (Figura 6.7). Selecione esta opção com um duplo clique.

Em seguida, na barra de ferramentas desta tela, selecione a opção “Network Configuration”. Caso não apareça nenhuma rede identificada, clique em “Search Network” (Figura 6.8).

Cap. 7 – Funções Especiais em CLPs Modicom-Telemecanique, Festo e Omron

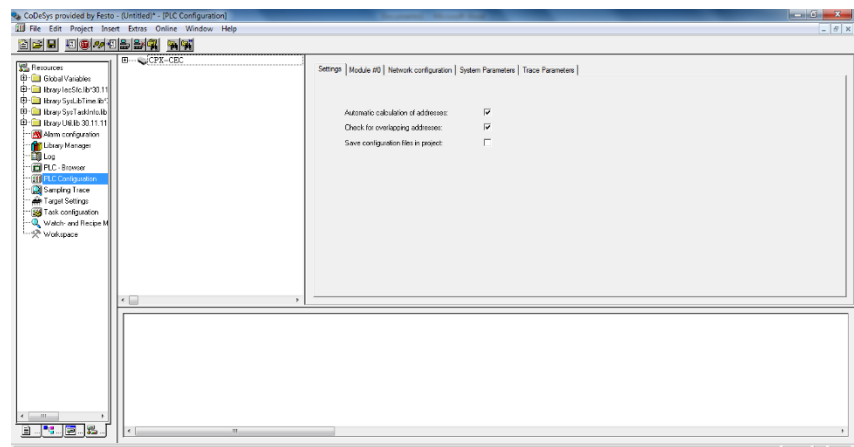


Figura 6.7 – Seleção de PLC Configuration

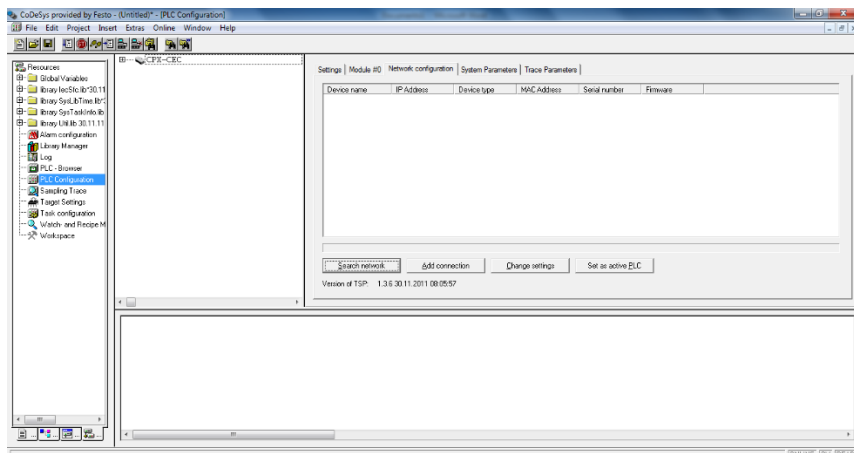


Figura 6.8 – Seleção de “Search Network”

Agora devem aparecer na tela todos os dispositivos (computadores) conectados na rede. Os computadores e seus respectivos números de IP são ilustrados na Figura 6.9.

Selecione o IP da bancada em que estiver trabalhando e “Set as Active PLC”. O IP configurado é destacado em azul como ilustrado nas Figuras 6.10 e 6.11.

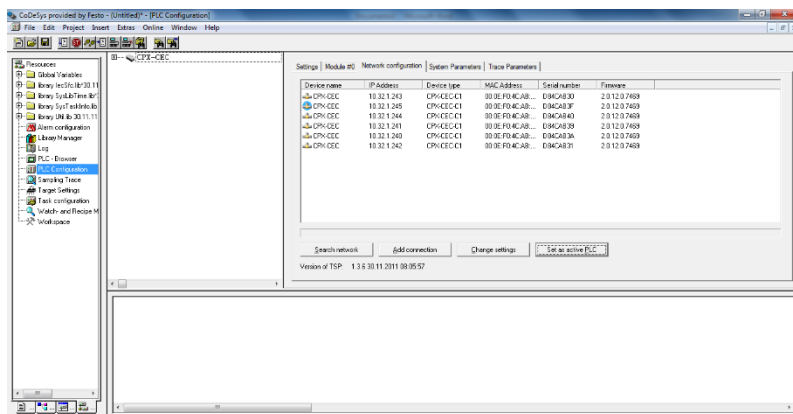


Figura 6.9 – Números dos IPs dos Computadores da Rede

Em seguida selecione CPX-CEC e IOModules, para configurar os módulos de IO que estão alocados no PLC e selecione “Scan Module Configuration” (Figura 6.12 e 6.13).

Cap. 7 – Funções Especiais em CLPs Modicom-Telemecanique, Festo e Omron

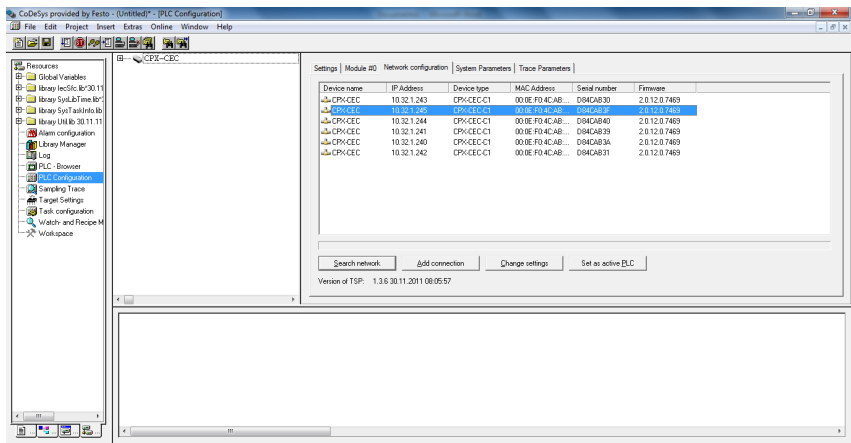


Figura 6.10 – Selecionando o PLC Ativo

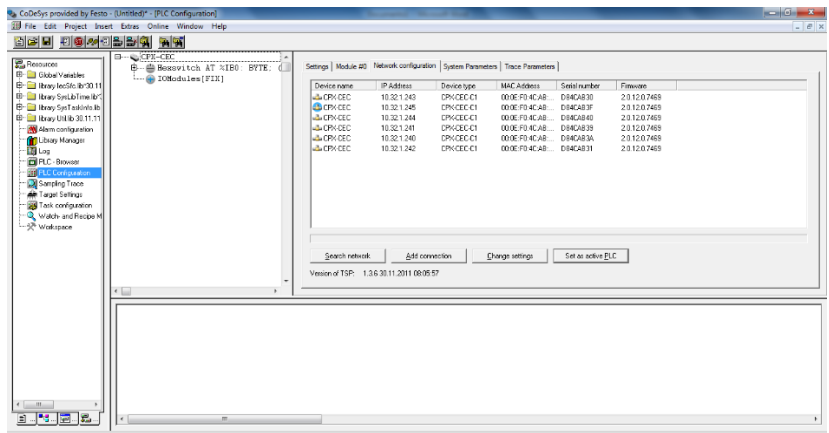


Figura 6.11 – Selecionando o PLC Ativo - Continuação

Cap. 7 – Funções Especiais em CLPs Modicom-Telemecanique, Festo e Omron

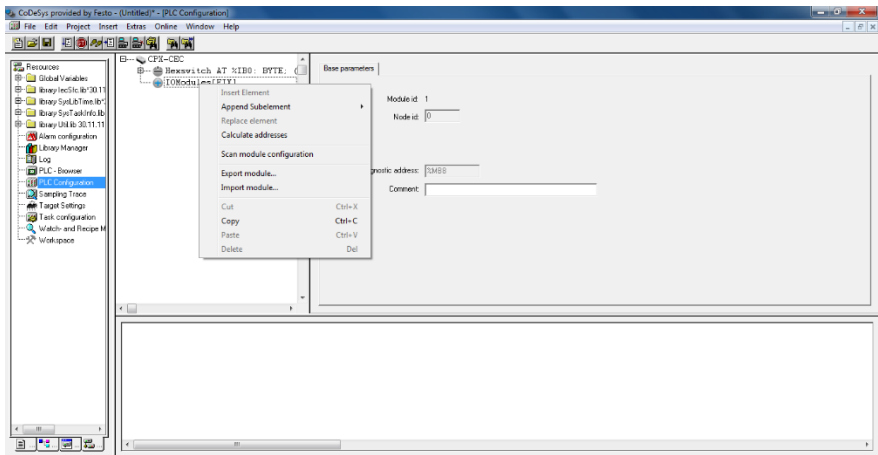


Figura 6.12 – Identificando e Selecionando os Módulos

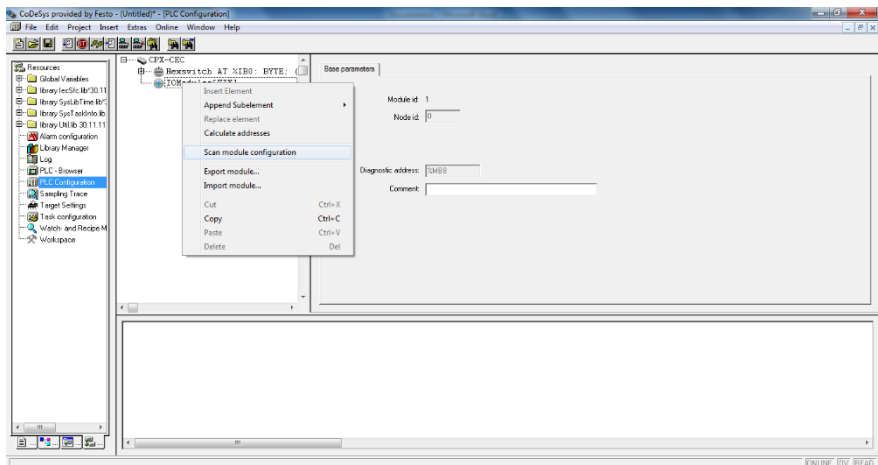


Figura 6.13 – Identificando e Selecionando os Módulos - Continuação

Após essa configuração é possível verificar os módulos de IO (Figuras 6.14 e 6.15). Note que temos dois módulos de IO, cada um com 8 entradas e 8 saídas. Selecionando cada módulo é possível identificar os

endereços de acesso a cada um deles. Por exemplo, o primeiro módulo tem os endereços de entrada acessados utilizando %IX2.0 a %IX2.7. Esses são os endereços que serão utilizados na programação quando quiser utilizar as entradas do primeiro cartão de IO. Os endereços de saída do primeiro cartão são %QX0.0 a %QX0.7. Para o segundo cartão, os endereços são: %IX4.0 a %IX4.7 e %QX2.0 a %QX2.7.

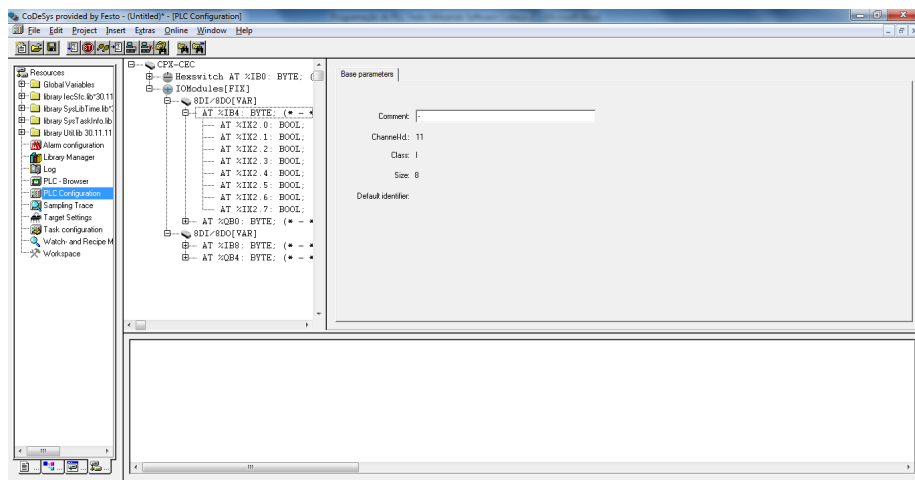


Figura 6.14 – Verificando os Endereços dos Módulos de IO

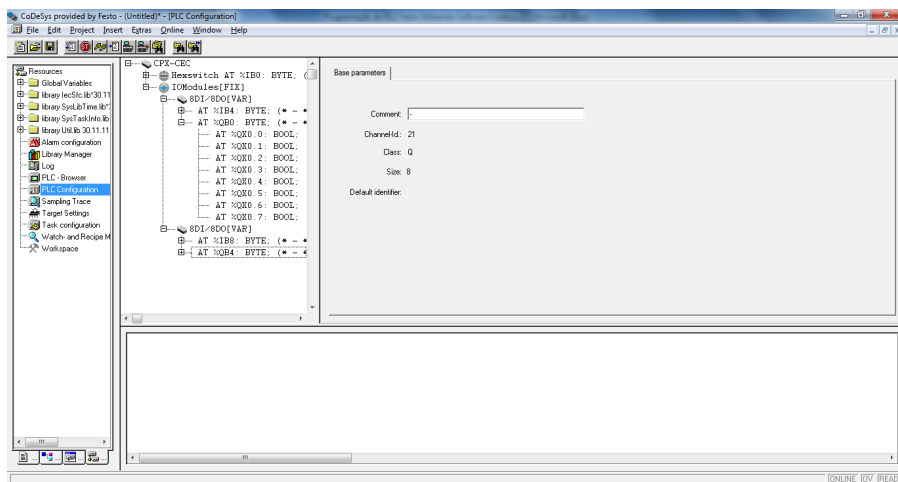


Figura 6.15 – Verificando os Endereços dos Módulos de IO - Continuação

Após essas configurações estamos prontos para programar.

Para ilustrar será apresentada na sequência um **programa simples em Ladder**.

Para realizar um projeto simples, primeiramente iremos retornar à tela de programação clicando no atalho POU's, na lateral inferior esquerda. Se a janela já estiver aberta, é só minimizar a tela de “Resources” e abrir a tela de programação (Figura 6.16). A tela de programação é apresentada na Figura 6.17.

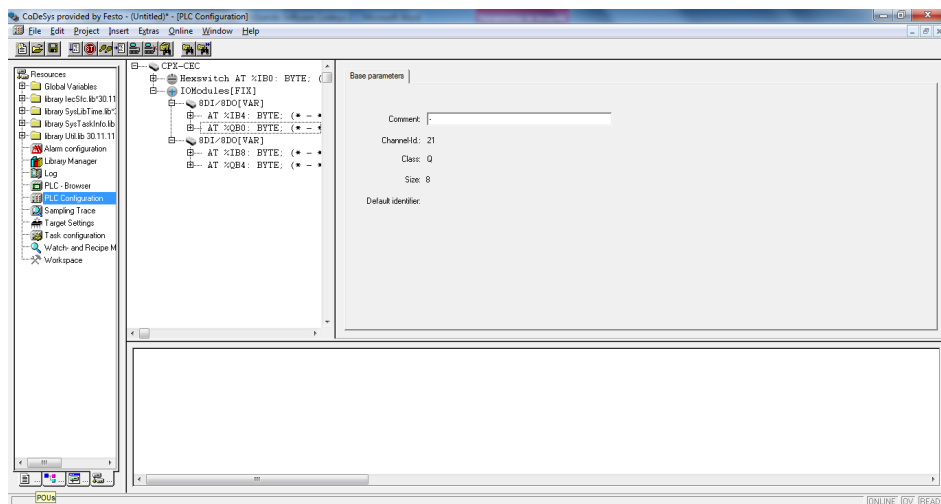


Figura 6.16 – Retornando à Tela de Programação

A barra de ferramentas superior apresenta um atalho para os elementos da programação em Ladder. Para testar a programação e a comunicação com o PLC realize um projeto simples como o ilustrado na Figura 6.18.

A cada elemento inserido, quando definimos o label da entrada ou saída (por exemplo, Botão1), abre-se uma janela para configurarmos a entrada correspondente (Figura 6.19). Note que o programa desenvolve a automática configuração das variáveis em Ladder mostrando o código em Lista de Instruções. Esse código não é o programa, mas apenas a configuração das variáveis.

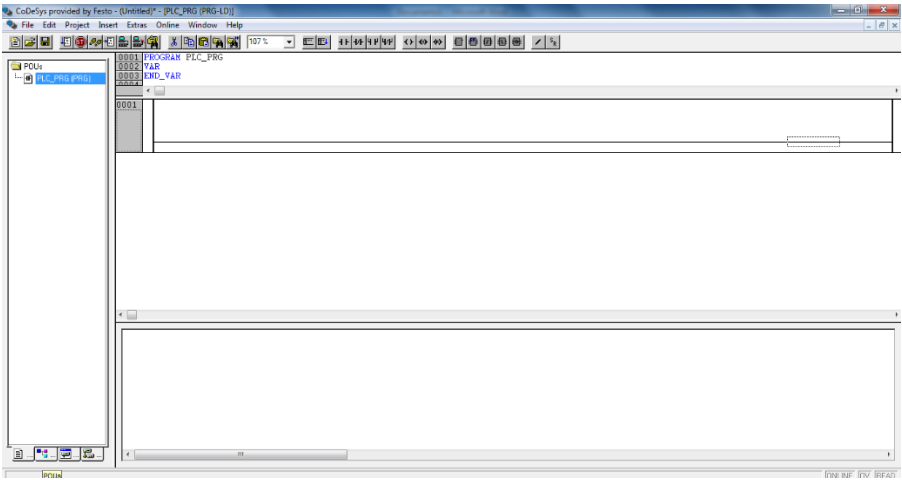


Figura 6.17 – Tela de Programação em Ladder

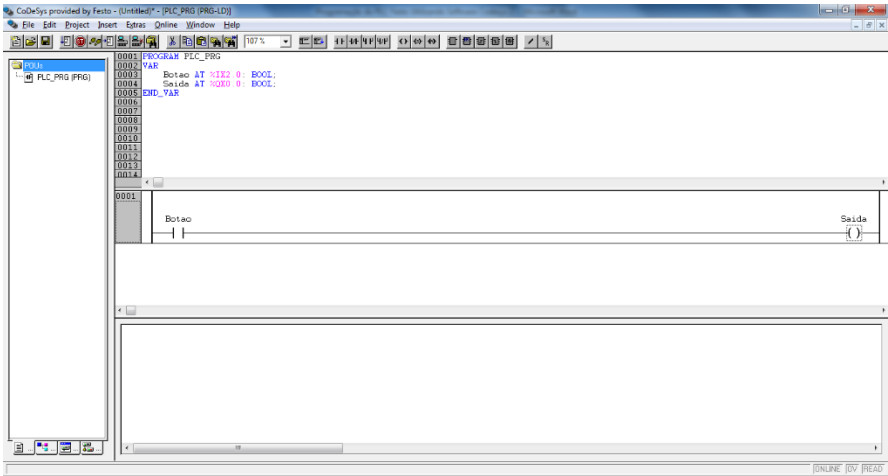


Figura 6.18 – Um Exemplo Simples de Programação em Ladder

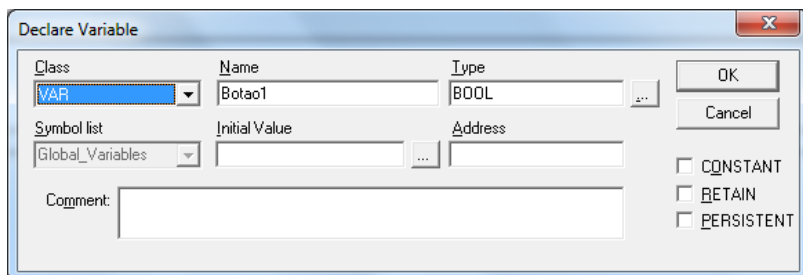


Figura 6.19 – Tela de Configuração de Entrada, Saída ou Variável

Para transferir o programa clique no atalho “Login” na barra de ferramentas. Em seguida clique em Run, para conseguir visualizar a operação no computador. Agora é possível testar o funcionamento conectando 24 V na entrada escolhida e verificando se a saída será comandada.

6.3 – Programação de Temporizadores e Contadores e Outras Funções no CLP Festo Usando CoDeSys.

Nesta seção serão apresentados os aspectos de programação do CLP Festo. Daremos ênfase à programação de temporizadores e contadores em Ladder do CLP utilizando o Software CoDeSys. Algumas aplicações típicas serão propostas para fixação dos conceitos. Para isso, também neste capítulo serão apresentados aspectos relativos às ferramentas que podem ser utilizadas no desenvolvimento das aplicações.

Temporizadores

Os temporizadores podem ser inseridos na programação utilizando a função Timer TON (Figura 6.20). Para isso o software deve estar desconectado (Logout).

Configure o Timer diretamente com o seu endereço (T0, T1, assim por diante) clicando sobre o temporizador. Assim, o Address não precisa ser configurado. Ajuste o valor de temporização (Figura 6.21). Exemplo: t#5s configura o tempo de 5 segundos. Esse exemplo pode ser transferido para o PLC e testado.

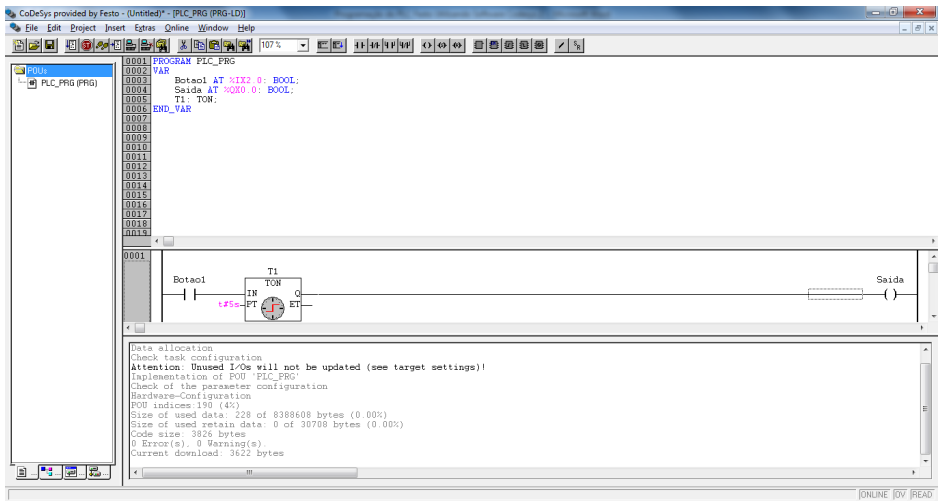


Figura 6.20 – Função Timer da Programação em Ladder no CoDeSys

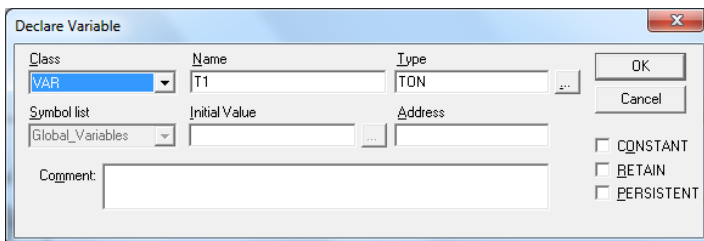


Figura 6.21 – Configuração do Timer da Programação em Ladder no CoDeSys

Contadores (CNT)

Os contadores podem ser inseridos na programação utilizando o atalho “Function Block” na barra de ferramentas (Figura 6.22).

Para isso, clique nesse atalho e selecione a função Counter para contagem decrescente (CTD – Counter Down), contagem crescente (CTU – Counter Up) ou para contagem crescente e decrescente (CTUD). Nesse exemplo estamos utilizando o CTD.

Configure o contador diretamente com o seu endereço (CTD0, CTD1, assim por diante) clicando sobre o contador. Assim, o Address não precisa ser configurado (Figura 6.23). Ajuste o valor de contagem (Process Value). Exemplo: 3. Configure também a entrada que irá resetar o contador (Figura 6.24). Exemplo: Início, configurada como entrada %IX4.0 ou %IX2.1 Por fim, caso queira monitorar os valores de contagem, configure a variável CONT como WORD (Figura 6.25). No exemplo final, apresentado na Figura 6.26 é também utilizada variável auxiliar AUX (bit de memória).

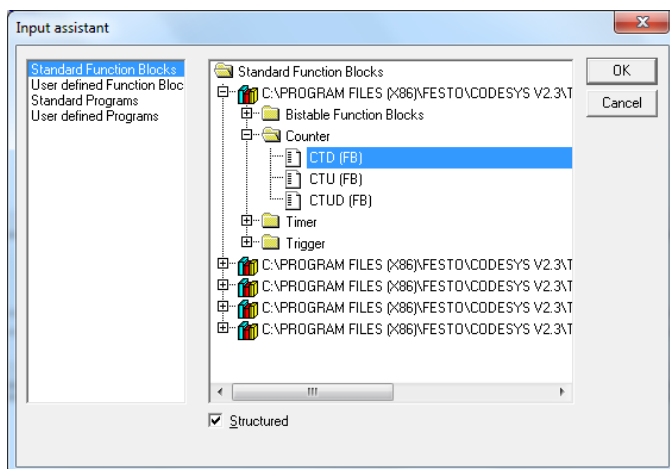


Figura 6.22 – Introdução do bloco de funções do contador na Programação em Ladder no CoDeSys

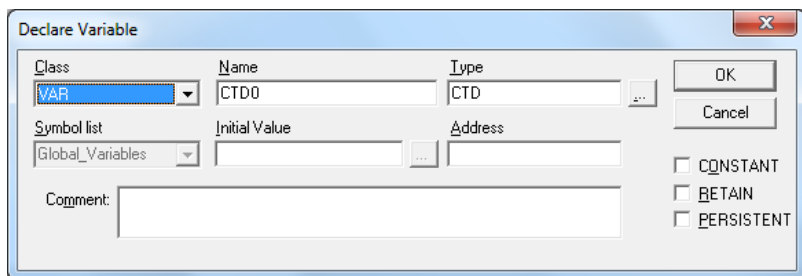


Figura 6.23 – Configuração do Contador da Programação em Ladder no CoDeSys

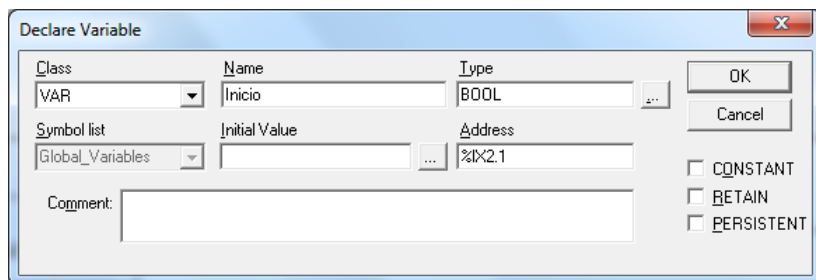


Figura 6.24 – Configuração da Entrada para Resetar o Contador

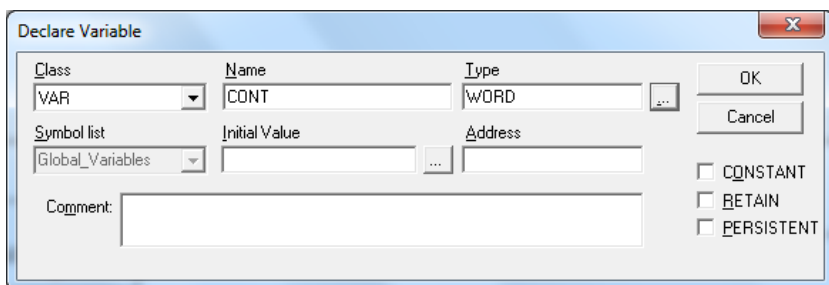


Figura 6.25 – Configuração da Entrada para Monitorar o Valor de Contagem

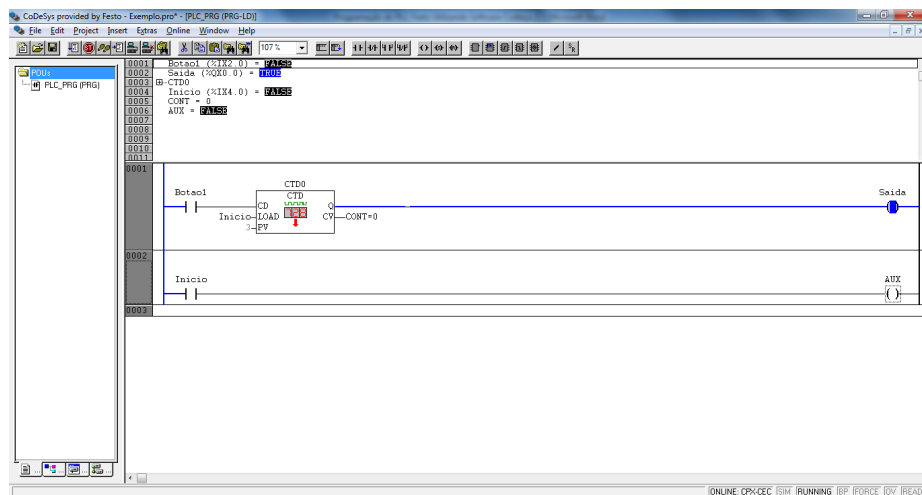


Figura 6.26 – Exemplo Simples de Utilização de Contador na Programação em Ladder no CoDeSys

6.4 – Exercícios de Fixação.

Visando fixar os conceitos de pneumática e de programação do CLP FEC Festo apresentados serão agora propostos alguns exemplos de aplicação. Sugere-se que o leitor utilize os conceitos apresentados até aqui para desenvolver programas em Ladder e STL que implementem estas aplicações.

Exercício 6.1 – Acionamento de um Atuador de Dupla Ação

a) Considere um atuador de dupla ação comandado pelo acionamento dos botões S_1 , S_2 e S_3 onde S_1 e S_2 são *push-buttons* (portanto quando acionados produzem pulsos) e S_3 apresenta intertravamento.

O controle da atuação e recuo da haste do cilindro é comandado através de uma válvula 5/2 vias, NF, acionamento por solenoide, retorno por mola.

Elabore um programa em Ladder e STL para acionar o atuador **enquanto** os botões S_1 e S_2 são mantidos acionados simultaneamente ou quando S_3 é acionado.

Apresente também o circuito pneumático e o diagrama de interconexões elétricas utilizados para a implementação.

b) Elabore um programa (em Ladder e/ou STL) para que, quando os botões S_1 e S_2 forem acionados simultaneamente pela primeira vez ou enquanto S_3 estiver acionado, o atuador seja acionado mantendo-se assim. O recuo do atuador deverá ocorrer pela repetição da ação (novo acionamento simultâneo dos botões S_1 e S_2 ou desacionamento de S_3).

Resumindo, deseja-se que, via software, os botões S_1 e S_2 (que não possuem intertravamento), apresentem retenção.

Exercício 6.2 – Sistema Automático de Furação de Peças

Considere um sistema automático para furação de peças (Figura 6.27) que executa o seguinte procedimento:

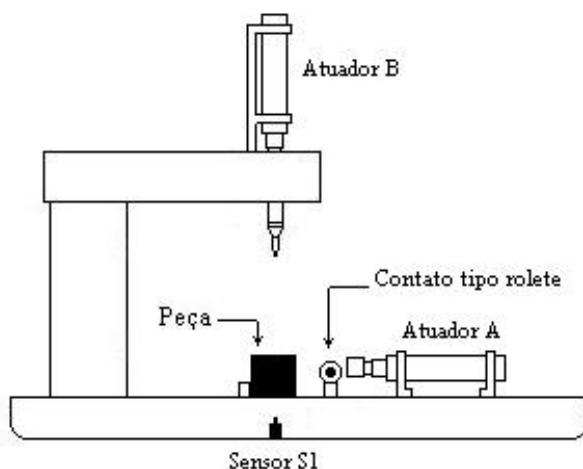


Figura 6.27 – Sistema Automático de Furação de Peças

No instante inicial, uma peça transportada através de uma esteira é colocada sobre uma base de fixação. Um botão S_1 (ou um sensor) é acionado, indicando que a peça está na posição adequada. O sinal é reconhecido pelo CLP e este provoca o acionamento do atuador A que é responsável pela fixação da peça. Então a haste do atuador A avança, acionando um contato NA (chave 1) acionado por rolete.

O atuador B, que realiza a furação é então automaticamente acionado. O tempo necessário para o avanço da haste do atuador B e furação da peça consiste de aproximadamente 2 segundos, sendo que o atuador B recua após este tempo.

O atuador A retorna à posição inicial, mas somente 2 segundos após o recuo do atuador B (por segurança). Observe então que o cilindro atuador A se mantém atuado durante um período total de 4 segundos.

O sistema fica então paralisado aguardando até que uma nova peça seja colocada na base de fixação (ou seja, o botão ou sensor seja novamente acionado).

Considere que os atuadores são ambos cilindros de dupla-ação, sendo acionados respectivamente pelas seguintes válvulas direcionais:

- Atuador A - Válvula 5/2 vias, NF, acionamento por solenoide, retorno por mola
- Atuador B - Válvula 5/2 vias, NF, acionamento por duplo solenoide

Elabore um programa em Ladder ou em STL para desenvolver a aplicação.

Apresente também o circuito pneumático e o diagrama de interconexões elétricas utilizados para a implementação.

OBS – Considere que a unidade de furação possui um sistema de amortecimento de final de curso que reduz a velocidade durante o processo de furação. Este efeito não afeta a programação.

Exercício 6.3 – Sistema de Identificação e Contagem de Peças

Considere um processo industrial (Figura 6.28) em que uma esteira acionada por um motor transporta peças de dois tamanhos (pequena e grande) que sensibilizam dois sensores (ótico, indutivo ou capacitivo).

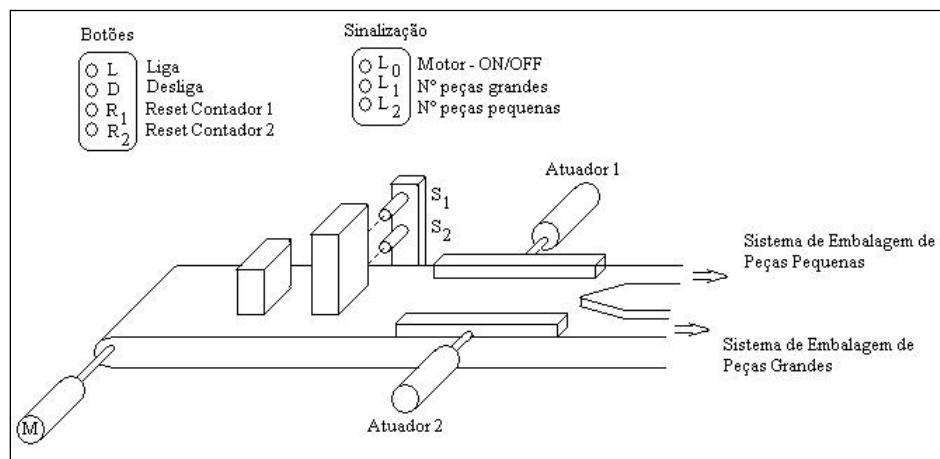


Figura 6.28 – Sistema de Identificação e Contagem de Peças

O objetivo deste projeto é fazer a detecção de peças grandes e pequenas, posicionar as peças para embalagem e fazer a contagem de peças.

O sistema é comandado por quatro botões, todos sem retenção e utiliza três LEDs para sinalização.

Quando o botão L é acionado o processo é inicializado e o motor é acionado sendo sinalizado através do LED L₀.

O processo pode ser interrompido em qualquer momento através do botão D. Neste caso, L₀ é apagado e os contadores são reinicializados.

A identificação do tipo de peça é feita pelos sensores de forma que, se a peça é grande todos os sensores são ativados, se a peça é pequena somente um sensor é ativado.

Se a peça é grande devem ser efetuados os seguintes procedimentos:

- Posicionar a peça para embalagem através de um atuador (atuador 1). O atuador 2 deve estar recuado.

- Incrementar o contador de peças. É importante lembrar que, para cada peça somente deve ser feito um incremento. Para garantir isto, considere que o tempo que os sensores mantêm atuados na passagem de uma peça grande é de no mínimo 0,5 segundos.

- Se 6 peças grandes forem computadas, o processo é paralisado, sendo a sinalização visível através do LED L_1 (é aceso). Isto paralisa o processo (desliga o motor e LED L_0) enquanto permite sinalizar para o operador para que substitua a caixa de embalagem. Para restabelecer o processo o operador deve acionar o botão de reconhecimento R_1 para que o processo seja reiniciado, resetando o contador de peças que tinha atingido o limite da contagem. Atenção: o botão R_1 somente tem efeito se a contagem limite foi atingida.

Se a peça é pequena:

- Atuador 1 recuado e atuador 2 acionado.
- Incrementar o contador de peças para contar 5 peças.
- Se a contagem é atingida sinalizar através do LED L_2 . Para restabelecer o processo o operador deve acionar o botão de reconhecimento R_2 .

OBS:

- Os sensores S_1 e S_2 são estrategicamente colocados de forma que, no caso de peças grandes, S_1 atua antes de S_2 (evitando falsa detecção de peça pequena).
- Todos os botões são do tipo *push-button*.
- Serão utilizadas duas válvulas direcionais alocadas como quiser:
Válvula 5/2 vias, NF, acionamento por solenoide, retorno por mola
Válvula 5/2 vias, NF, acionamento por duplo solenoide

Elabore o programa em Ladder ou STL para implementar esta aplicação.

7 – FUNÇÕES ESPECIAIS EM CLPS MODICOM-TELEMECANIQUE (SCHNEIDER ELECTRIC), FESTO E OMRON

No desenvolvimento de aplicações mais elaboradas algumas funções especiais podem facilitar em muito a programação dos controladores lógicos programáveis. Existe uma grande diversidade de funções especiais disponíveis na maioria dos programas de desenvolvimento, podendo ser utilizadas em várias linguagens. Podemos citar como exemplos algumas funções:

- Registradores.
- Comparadores.
- Operações Aritméticas.
- Função PWM.
- Controle PID.

Neste capítulo serão apresentados os aspectos de programação dessas funções especiais. Como referência serão considerados modelos de CLPs da Schneider (Modicom-Telemecanique), Festo e Omron.

7.1 – Programação do PLC Modicom-Telemecanique no software PL7.

A programação do CLP Modicom – Telemecanique® é efetuada diretamente através do software PL7 instalado num PC (*Personal Computer*).

A título de ilustração considera-se a utilização do modelo com processador TSX3722 V3.3 que contém os seguintes módulos:

- Módulo analógico, contendo 8 canais de entrada analógica e 1 canal de saída, todos de 8 bits com ciclo de acesso de 4ms x número de canais utilizados.
- 2 Módulos de contadores rápidos. Módulo da CPU.

- Módulo duplo de E/S, contendo 16 entradas digitais (24 V), e 12 saídas digitais a relé (portanto podendo acionar diretamente dispositivos externos).
- Espaço para conexão de outros módulos de entradas ou saída.
- Módulo da CPU.
- Módulo duplo de E/S, contendo 16 entradas digitais (24 V), e 12 saídas digitais a relé (portanto podendo acionar diretamente dispositivos externos).
- Espaço para conexão de outros módulos de entradas ou saída.

A configuração do módulo no software PL7 é realizada como demonstrado na Figura 7.1

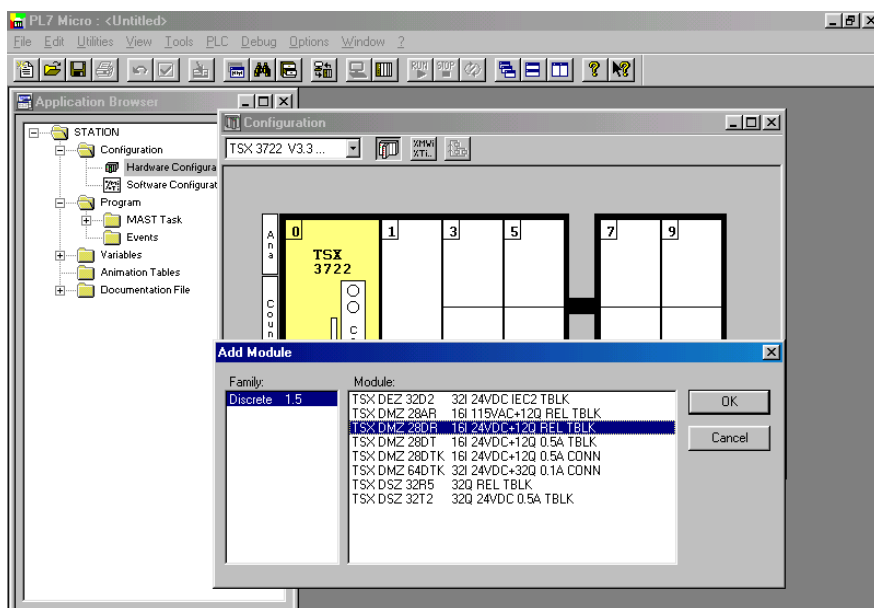


Figura 7.1 – Configuração de hardware do CLP Modicom-Telemecanique®

O modelo apresenta as seguintes funções disponíveis:

- Timers (TM) – Até 64 temporizadores.
- Series 7 (T) – Incluídos dentre os 64 temporizadores.
- Monoestável (MN) – 8 monoestáveis.
- Counters (C) – 32 contadores.
- Registradores de 16 bits (R) – R₀ a R₃. Podem ser alterados para 8 bits.

O software PL7 permite introduzir além dos contatos e bobinas NA e NF convencionais as seguintes funções: bobinas set e reset, contato de acionamento na borda de subida, contato de acionamento na borda de descida, chamada e retorno de sub-rotina, temporizadores (TON, TOF e TP), contador (crescente & decrescente) e temporizador Série 7. Algumas das funções mais importantes do PLC são detalhadas a seguir:

Temporizador TON (Temporizador na Energização)

A introdução dos temporizadores é efetuada através da função *Graphic Function* acessada através dos botões de função na barra de ferramentas inferior ou usando *F7*.

Configurações a serem efetuadas:

- Endereço (Ex: %TM0, %TM1, ...)
- MODE: TON
- Base de tempo (1 min, 10ms, 1 s, etc.)
- Timer Preset
- Modif: Y (Permite modificação On-Line)

A Figura 7.2 ilustra um exemplo de programação usando temporizador TON no software PL7.

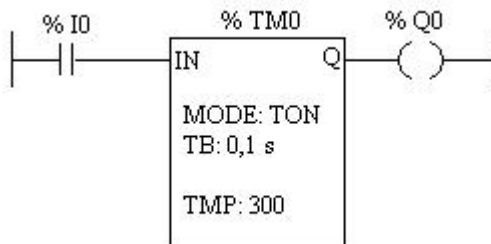


Figura 7.2 – Implementação do Temporizador TON no Software PL7

O parâmetro TM.V indica o valor corrente do temporizador, que aparece após a introdução do temporizador, depois que o CLP for conectado e estiver rodando (RUN).

A saída do temporizador TON somente é habilitada se a entrada do temporizador é energizada durante o tempo ajustado. Quando a entrada cai a zero, a contagem é resetada. A Figura 7.3 mostra o comportamento da saída com base na entrada.

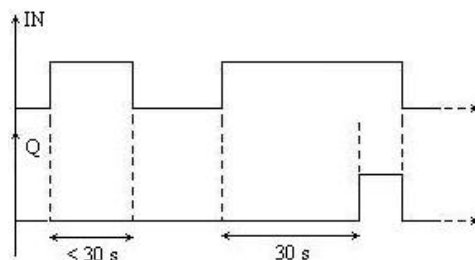


Figura 7.3 – Diagrama de Tempos do Temporizador TON
(CLP Modicom – Telemecanique®)

Temporizador TOF (Temporizador na Desenergização)

Apresenta o comportamento similar ao TON. Contudo a saída somente é habilitada se a entrada do temporizador é mantida desabilitada durante o tempo ajustado. Quando a entrada vai para um, a contagem é resetada, conforme ilustrado na Figura 7.4.

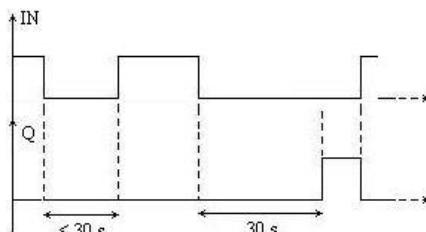


Figura 7.4 – Diagrama de Tempos do Temporizador TOFF (CLP Modicom – Telemecanique®)

Temporizador TP (Temporizador de Pulso)

Apresenta o comportamento similar ao de um monoestável conforme apresentado anteriormente. Ao aplicar um pulso na entrada a saída será imediatamente habilitada, mantendo-se assim durante o tempo ajustado, conforme ilustrado na Figura 7.5.

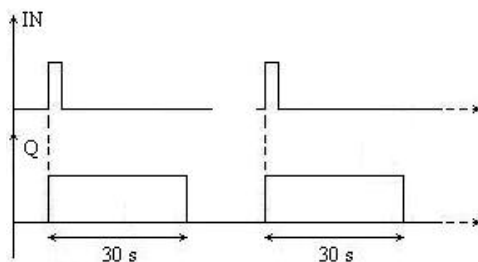


Figura 7.5 – Diagrama de Tempos do Temporizador TP (CLP Modicom – Telemecanique®)

Contador

A introdução do contador é também efetuada através da função *Graphic Function* acessada através dos botões de função na barra de ferramentas inferior ou usando *F7*.

Configurações a serem efetuadas:

- Endereço (Ex: %C0, %C1, ...)
- Contador Preset
- Modif: Y (Permite modificação On-Line)

O parâmetro C.V indica o valor corrente do contador, que aparece após a introdução do contador, depois que o CLP for conectado e estiver rodando (RUN).

O contador, apresentado na Figura 7.6, apresenta 4 entradas e 3 saídas que podem ser utilizados pelo usuário conforme o seu interesse.

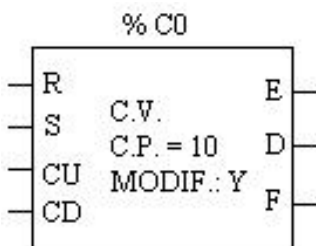


Figura 7.6 – Configuração Básica do Contador
(CLP Modicom – Telemecanique®)

As entradas do contador são:

- R – Entrada de reset. Reseta a contagem quando esta entrada estiver habilitada.
- S – Entrada de set. Seta a contagem (valor 9999) quando esta entrada estiver habilitada.
- CU – Entrada do contador crescente. Permite através de pulsos incrementar a contagem no contador.
- CD – Entrada do contador decrescente. Permite através de pulsos decrementar a contagem no contador.

As saídas do contador são:

- D – Habilita a saída quando o valor corrente do contador atinge o valor de preset ajustado.
- E – Habilita a saída quando o valor corrente do contador cai de 0 para 9999.
- F – Habilita a saída quando o valor corrente do contador vai de 9999 para 0.

Temporizador Series 7 (T)

O temporizador série 7 (Figura 7.7) pode também ser introduzido através da função *Graphic Function*. Neste caso, é necessário voltar à configuração de hardware para selecionar pelo menos 1 temporizador serie 7, que corresponde ao temporizador do primeiro endereço (%T0). Este temporizador corresponde a um contador de tempo decrescente.

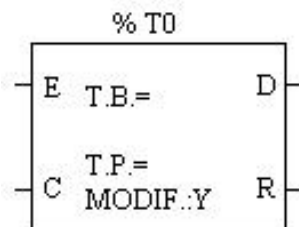


Figura 7.7 – Configuração Básica do Temporizador Série 7
(CLP Modicom – Telemecanique®)

As entradas do contador são:

- E – Enable. Habilita o contador quando esta entrada estiver habilitada. Na borda de subida, ajusta o valor corrente do contador de tempo no seu valor de preset.
- C – Entrada do contador. Permite através de pulsos decrementar a contagem no contador.

As saídas do contador são:

- D – Habilita a saída quando o valor corrente do contador atinge o valor 0.
- R – Corresponde ao complemento de D. Mantém a saída habilitada enquanto a contagem não for atingida. Quando atinge o valor 0 desabilita a saída.

7.2 – Registradores.

Os registradores são dispositivos utilizados para armazenamento temporário de dados, operações de transferência e realização de operações. Existem várias formas para realizar procedimentos e programação de registradores nos CLPs.

Abaixo seguem algumas informações sobre o uso de registradores nos CLPs Modicom Telemecanique, FESTO e Omron.

Na programação dos CLPs Modicom-Telemecanique existem disponíveis um grande número de registradores de 8 bits (*Bytes*), 16 bits (*Words*) e 32 bits (*Double-Words*). Para utilizar as *Words* por exemplo, basta utilizar as entradas ou saídas representadas pelo padrão %MWX onde X = 0, 1, 2, 3, O número de registradores disponíveis no CLP Modicom-Telemecanique pode ser visualizado no menu de configuração de hardware.

No CLP FESTO (modelo FEC Standard) há 256 registradores de 16 bits disponíveis com endereçamento R0 até R255, sendo que R0 a R127 são registradores retentivos e R128 a R255 são não-retentivos. Os registradores podem ser referenciados pelo padrão RWX onde X = 0, 1, 2, 3, Como os registradores são de 16 bits, podem assumir valores V0 a V65535.

Para carregar um valor no registrador do CLP FEC devemos escolher a linha que receberá o comando, em seguida acionar o botão de atalho operação Multi-bit e selecionar o item TO. Este comando é na verdade o LOAD TO, ou seja, carregar para, o qual permite ajustar um valor no registrador. Do lado esquerdo da caixa criada deverá ser colocado o valor (V00000 a V65535) ou o objeto que deverá ser transferido para o registrador. O objeto pode ser o valor proveniente de uma entrada/saída analógica (IW ou OW) para os modelos que as possuem, *Timer Preset* (TP), *Timer Word* (TW), *Counter Word* (CW), etc. No lado direito define-se o registrador.

A mesma ação de carregar um valor no registrador pode ser desenvolvida também em Lista de Instruções, como no exemplo:

IF	I1.0
THEN LOAD	V65535
TO	RW0

7.3 – Comparadores.

Na comparação são utilizados dois operandos de 16 bits que podem ser indexados ou configurados imediatamente.

A habilitação da comparação é feita pela entrada *EN (Enable)* quando no estado 1.

O resultado da comparação pode ser obtido pelo monitoramento de 4 saídas. São elas:

- > ligada quando o conteúdo do primeiro operando é maior que do segundo;

- = ligada quando o conteúdo do primeiro operando é igual ao do segundo;

- < ligada quando o conteúdo do primeiro operando é menor que do segundo;

- <> ligada quando o conteúdo do primeiro operando é diferente ao do segundo.

Um exemplo desenvolvido na programação em Ladder do CLP Modicom-Telemecanique é ilustrado na Figura 7.8, onde efetua-se a comparação de um sinal de entrada analógica e um valor constante (5000). No exemplo, a condição para que a comparação seja efetuada é que a entrada %I1.0 esteja habilitada. Se o sinal da entrada analógica tem valor maior ou igual a 5000 liga-se a saída %Q2.0 e desliga-se a saída %Q2.1. A lógica inversa ocorre quando o sinal de entrada for menor que 5000. Alternativamente a entrada analógica %IW0.5 poderia ser alterada para uma memória (registrador), por exemplo %MW20.

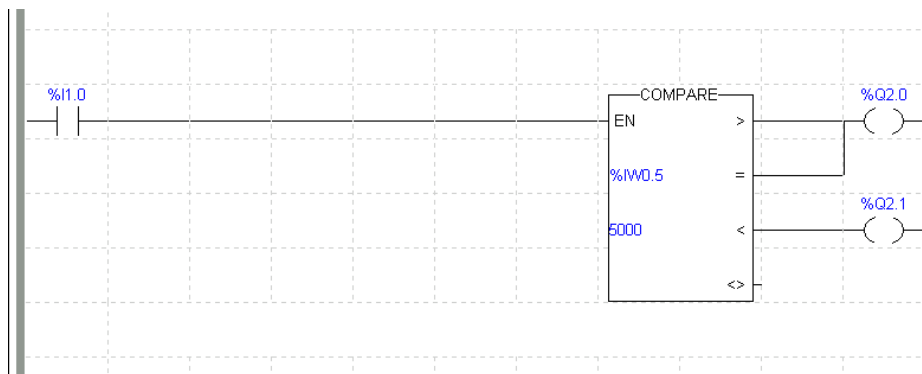


Figura 7.8 – Utilização de funções de comparação no software PL7

Na programação do CLP FEC em Ladder, a função de comparação pode ser introduzida acionando o botão de atalho inserir Comparadores e em seguida selecionar a função desejada, ou seja, Maior, Menor, Igual, Diferente, Menor ou Igual ou Maior Igual, representados respectivamente pelos símbolos >, <, =, <>, <=, >=. É possível fazer a comparação do valor atual de um registrador com qualquer outro objeto (IW, OW, TW, TP, valores fixos, etc.).

Em Lista de Instruções (STL) utiliza diretamente os símbolos na programação, como no exemplo:

IF		CW0
	>	V535
THEN	SET	O2.0
JMP	TO	...

No CLP Omron, a programação em Ladder da função de comparação pode ser realizada como ilustrado na Figura 7.9. Note que no exemplo, compara-se o valor instantâneo de um temporizador (T0) com as constantes 100 e 150, para comandar a saída 1.00.

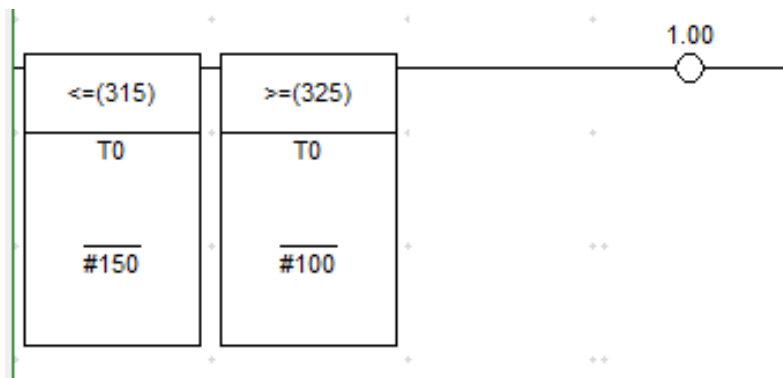


Figura 7.9 – Utilização de funções de comparação no software CX-Programmer

Outra forma de realizar a função é ilustrada na Figura 7.10. No exemplo utiliza-se um temporizador habilitado quando a entrada 0.00 é acionada. Estando o temporizador 0.00 ele começa a contar tempo decrementando a contagem a partir do valor do *preset*. Para detectar um determinado valor de temporização instantâneo podemos utilizar uma função de comparação e após realizar o teste de um flag (por exemplo, o flag de *Zero* ou o flag de *Carry*). No exemplo da Figura 7.10 o valor instantâneo de temporização no timer T0 é comparado com 100 s para verificar se esse valor de contagem foi atingido. Considerando que a contagem começa em 150 s, podemos concluir que após 50 s (ou seja, 150 s - 100 s) o resultado da comparação irá apresentar resultado zero. Assim, o teste do flag P_EQ (flag de Zero) após a comparação nos permitirá detectar quando o tempo de 50 s for atingido o qual ocorrerá quando P_EQ = 1 o que permitirá ligar a saída 1.00. Testes similares podem ser realizados para detectar outros limiares de tempo. Adicionalmente o uso do flag P_CF (flag de carry) permitirá detectar quando o tempo atingido é menor que o valor de comparação (P_CF = 1) ou quando o valor atingido é maior que o valor de comparação (P_CF = 0).

Para realizar operação similar podem ser utilizadas instruções textuais na linguagem ST (*Structured Text*) no CLP Omron. Um exemplo é ilustrado na Figura 7.11.

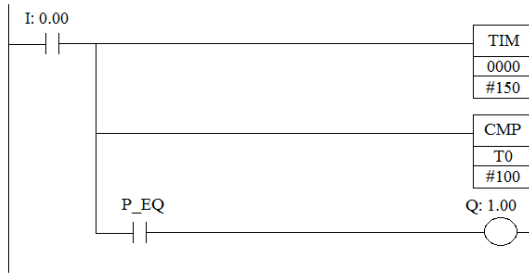


Figura 7.10 – Utilização de funções de comparação e *flag* no software CX-Programmer

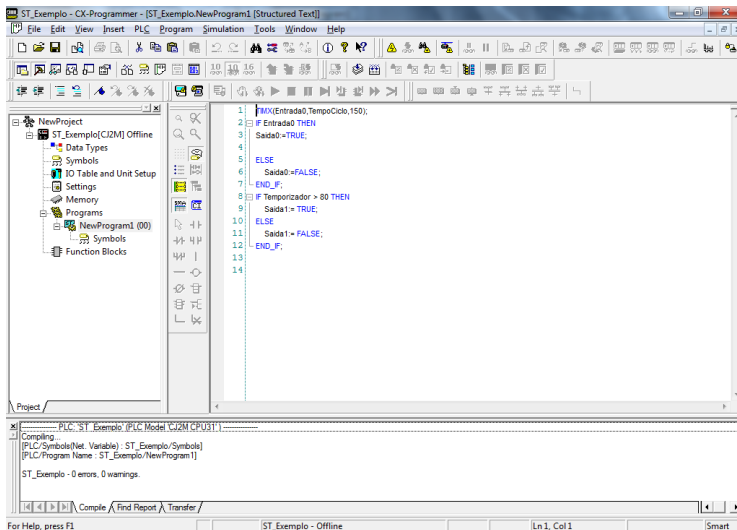


Figura 7.11 – Exemplo de programação em linguagem ST no software CX-Programmer

Um exemplo de código para definir o temporizador (TMX) e configurá-lo, bem como testar o valor de comparação dentro de uma estrutura IF - THEN - ELSE é apresentada na Figura 7.12.

```

1 | TMX(Entrada0,TempoCiclo,150);
2 | IF Entrada0 THEN
3 |   Saida0:=TRUE;
4 |
5 | ELSE
6 |   Saida0:=FALSE;
7 | END_IF;
8 | IF Temporizador > 80 THEN
9 |   Saida1:= TRUE;
10 | ELSE
11 |   Saida1:= FALSE;
12 | END_IF;

```

Figura 7.12 – Exemplo de código em linguagem ST no software CX-Programmer

É importante lembrar que para desenvolver aplicações em texto estruturado (ST) é necessário definir todos os símbolos utilizados, como ilustrado no exemplo da Figura 7.12 que apresenta os símbolos definidos para o exemplo da Figura 7.13.

Entrada0	BOOL	0.00	Main Rack :...	In
Saida0	BOOL	1.00	Main Rack :...	Out
TempoCiclo	TIMER	T0000		Work
Saida1	BOOL	1.01	Main Rack :...	Out
Temporizador	INT	T0		Work

Figura 7.13 – Símbolos definidos para o exemplo da Figura 7.12

7.4 – Operações Aritméticas.

No CLP Modicom-Telemecanique é possível efetuar as operações de adição, subtração, Multiplicação e Divisão entre *words* (16 bits) e *double-words* (32 bits). Para isto são utilizados os operadores +, -, * e / conforme a seguinte notação:

Op1:=Op2 Operador Op3

onde Op1, Op2 e Op3 são operandos constituídos de *words* ou *double-words*.

Geralmente utilizam-se operandos indexados tais como *words* %MW0, %MW1, %MW2, etc., e *double words* %MD0, %MD1, %MD2, etc.

Para adicionar as operações na programação em Ladder utiliza-se o atalho *OPERATE* ou a tecla *F3*.

Exemplo: Na Figura 7.14 a entrada %I1.0 habilita a execução de duas operações: uma soma entre um *word* em memória e uma constante e uma multiplicação entre *words* em memória.

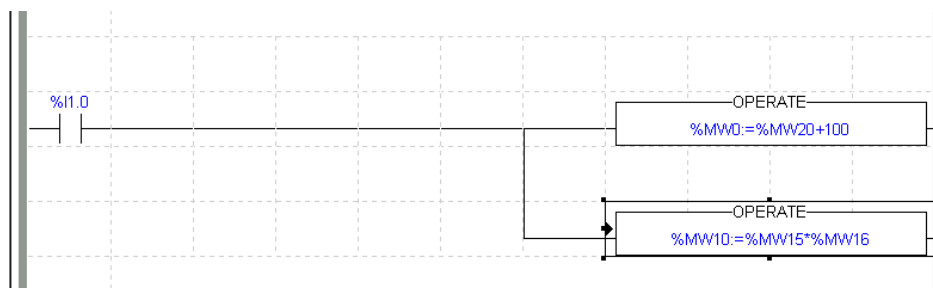


Figura 7.14 – Utilização de funções aritméticas no software PL7

No CLP FEC, as únicas operações aritméticas possíveis são o incremento e o decremento. Neste caso utilizam-se bobinas especiais de incremento e decremento, tal como já foi mostrado anteriormente para contadores. É possível incrementar objetos como registradores (R), temporizadores (T) e contadores (C). Por exemplo, para incrementar o contador registrador R0, basta utilizar bobina de incremento com esta identificação. Lembre-se que para incremento ou decremento em registrador não se utiliza o padrão RWX, mas RX mesmo, onde X = 0, 1, 2,

7.5 – Algumas Funções Especiais no PLC Modicom-Telemecanique.

Entradas Analógicas

O processador TSX3722 incorpora conversor AD com 8 canais de entrada analógica respectivamente endereçados por IW0.2, IW0.3, IW0.4, IW0.5, IW0.6, IW0.7, IW0.8 e IW0.9.

Dependendo do tipo de módulo, cada canal pode ser configurado para trabalhar com nas seguintes situações:

- medições elétricas (± 10 V, 0..10 V, 0..5 V, 1..5 V, 0..20 mA ou 4..20 mA),
- termo acoplamento (tipo B, E, J, K, L, N, R, S, T e U)
- medição de temperatura (Pt100 ou Ni1000).

No caso do protótipo do laboratório (processador TSX3722) somente é possível efetuar configurações de medições elétricas (0..10 V, 0..20 mA ou 4..20 mA). Para fazer estas configurações devemos utilizar o menu *Hardware Configuration* e clicando sobre o bloco analógico (*Ana*) podemos configurar o canal (2 a 9), a faixa de valores de medição (*range*) a escala e filtros.

A escala associa os máximos e mínimos valores de medição aos correspondentes valores de 16 bits medidos no conversor AD, sendo o

máximo valor correspondente a 10000 e o mínimo valor correspondente a 0. Por exemplo, se configurarmos *range* de ± 10 V, então uma medição de -10 V resulta numa leitura de valor 0 no registrador de saída do conversor AD. De forma similar, para uma medição de $+10$ V resulta numa leitura de valor 10000 no registrador de saída do conversor AD.

Os filtros são configurados pelo usuário conforme as características apresentadas na Tabela 7.1.

Valor Selecionado	Nível Requerido	Eficiência do Filtro (%)	Tempo de Resposta do Filtro	Frequência de Corte (kHz)
0	Sem filtro	0,0	0 ms	---
1	Baixo nível de Filtragem	75,0	111 ms	1,431
2		87,5	240 ms	0,664
3	Médio nível de Filtragem	93,7	496 ms	0,321
4		96,9	1,01 s	0,158
5	Alto nível de Filtragem	98,6	2,03 s	0,078
6		99,2	4,08 s	0,039

Tabela 7.1 – Configuração de Filtros no CLP Modicom – Telemecanique

Na bancada de laboratório encontram-se disponíveis as entradas I0.2, I0.3, I0.4 e I0.5, sendo que a primeira delas pode ser controlada manualmente por meio de um potenciômetro.

Saída Analógica

O processador TSX3722 incorpora conversor DA com 2 canais de saída analógica.

Dependendo do tipo de módulo, cada canal pode ser configurado para trabalhar com ± 10 V, 0..10V, 0..20 mA ou 4..20 mA, contudo no módulo TSX3722 somente um canal de saída está disponível, que trabalha com 0..10V e sendo endereçado por QW0.10. Na bancada de laboratório esta saída está ligada em um amperímetro.

Controle PID (Proporcional, Integral e Derivativo)

O controle de processos é uma das áreas de grande aplicação de controladores lógicos programáveis. Os controladores mais modernos incorporam a possibilidade de introdução de controladores PID digitais de maneira relativamente fácil.

A Figura 7.15 ilustra a introdução de um controlador PID por meio de linguagem ladder. Para inseri-lo utiliza-se o atalho *TEXT FUNCTION* ou a tecla *F8*. Selecionando a função *Process Control* e *PID* podemos fazer as primeiras configurações, ou seja:

- *TAG* – identificação do PID – nome definido pelo usuário;
- *UNIT* – Unidade a ser controlada (graus, rpm, etc.);
- *PV* – *Process Value* – identificação da variável que está sendo controlada, sendo medida através de um sensor e enviada para a entrada do controlador; pode ser uma entrada analógica ou o conteúdo de uma memória (exemplo: %IW0.2 ou %MW0.2);
- *OUT* – identificação da saída do controlador que pode ser uma memória ou uma saída analógica (exemplo: %QW0.10);
- *AUTO* – bit de seleção do modo de operação: manual (0) ou automática (1);
- *PARA* – identificação dos endereços dos parâmetros do controlador - 23 words (exemplo: %MW0 a %MW23, %MW20 a %MW43).

O formato final do controlador PID fica:

PID(TAG, UNIT, PV, OUT, AUTO, PARA)

A Figura 7.15 apresenta um exemplo de aplicação, sendo que nesse caso foi utilizada a configuração do PID como:

PID('PID1', 'Tensao', %IW0.2, %QW0.10, 1, %MW20:43)

Note que no exemplo foi considerando um controle PID automático para controlar a tensão de um sistema. Foi utilizado como medição o sinal PV proveniente de um potenciômetro (sensor) conectado na entrada %IW0.2 mas poderia ser utilizado um sinal proveniente de memória. O sinal de controle é enviado para a saída %QW0.10, mas também poderia ser enviada para uma memória e a partir desta informação produzir um sinal PWM (função que será descrita a seguir).

Os parâmetros de configuração são alocados nos endereços %MW20 a %MW43 e podem ser ajustados através da entrada de dados utilizando o *Application Browser* e criando uma tabela em *Animation Table*.

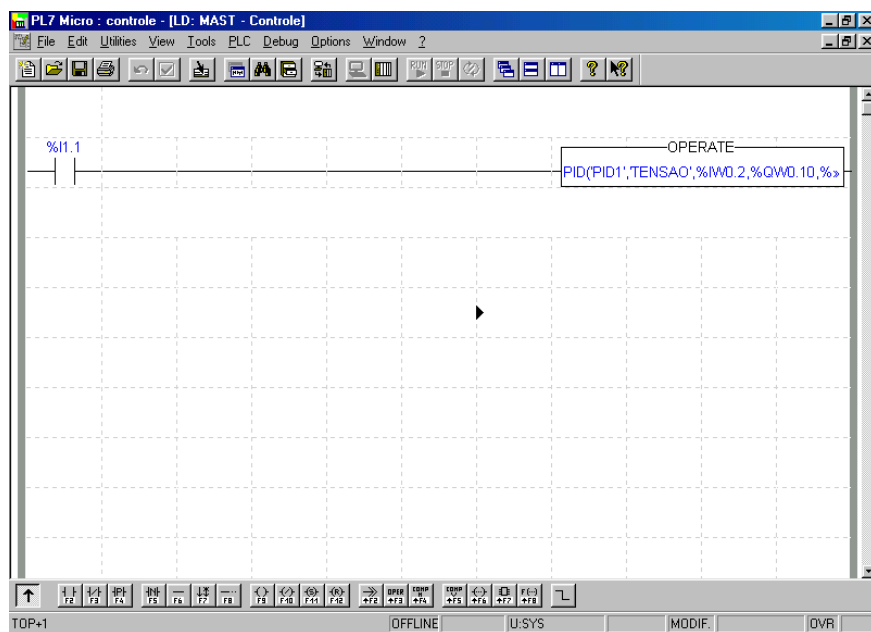


Figura 7.15 – Exemplo de Programação do Controlador PID (CLP Modicom – Telemecanique®)

A lista de configurações dos parâmetros do controlador PID são apresentados na Tabela 7.2. Nesta lista *i* representa o endereço inicial do pacote de 23 words utilizado para configuração de parâmetros (exemplo: *i* = 20 para seleção %MW20 a %MW43).

A Figura 7.16 apresenta uma tabela típica para ajuste dos parâmetros do controlador PID. Para entrar com os parâmetros é necessário entrar com todos os endereços e ajustar os valores desejados em *Current value* com o programa rodando.

Função PWM

Em vários sistemas de controle o sinal de controle produzido é convertido para um formato digital para acionar os dispositivos externos. Em sistemas de controle em eventos discretos utiliza-se o conceito de PWM (*pulse width modulation*) onde o sinal de controle variável é convertido num sinal digital com largura de pulso variável.

As principais características de um sinal PWM são:

- onda quadrada com amplitudes fixas (sinal digital oscilando entre 0 V e +Vcc);
- frequência fixa;
- variação do *duty cycle*, ou seja, do período em que o sinal apresenta nível lógico 1 em relação ao período total; ou ainda, em outras palavras, variação da largura de pulso.

Endereço ou Posição de Memória	Parâmetro	Função
%MWi	SP	Entrada de referência (setpoint)
%MW(i+1)	OUT_MAN	Controle manual
%MW(i+2)	KP	Ganho proporcional em série (<i>default</i> = 100); sinalizado de –10000 a +1000; o sinal indica o sentido direto ou reverso
%MW(i+3)	TI	Constante de tempo de integração em unidade 1/10 s (<i>default</i> = 0); valores entre 0 e 20000
%MW(i+4)	TD	Constante de tempo derivativa em 1/10 s (<i>default</i> = 0); valores entre 0 e 10000
%MW(i+5)	TS	Tempo de amostragem em 1/10 s; valores entre 10ms e 5 min 20s
%MW(i+6)	OUT_MAX	Limite máximo do sinal de controle
%MW(i+7)	OUT_MIN	Limite mínimo do sinal de controle
%MW(i+8) – bit 0	PV_DEV	Seleção de ação derivativa em medição (0) ou desvio (1)
%MW(i+1) – bit 8	DEVAL_MMI	Bit de inibição da função <i>man-machine interface</i> (interface homem-máquina para controle PID)
%MW(i+9)	PV_SUP	Limite superior de medição (<i>double-word</i>)
%MW(i+11)	PV_INF	Limite inferior de medição (<i>double-word</i>)
%MW(i+13)	PV_MMI	Imagem da medição do operador (<i>double-word</i>)
%MW(i+15)	SP_MMI	Setpoint do operador (<i>double-word</i>)

Tabela 7.2 – Configuração dos Parâmetros do Controlador PID (CLP Modicom – Telemecanique®)

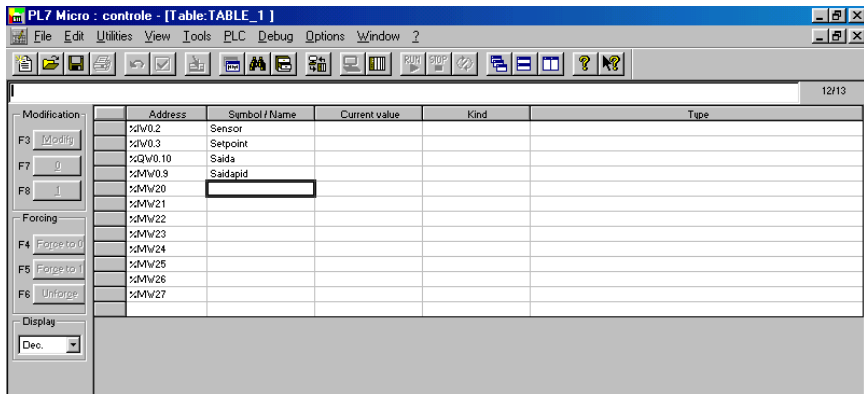


Figura 7.16 – Tabela de Animação (*Animation Table*) para Ajuste dos Parâmetros do Controlador PID (CLP Modicom – Telemecanique®)

Para inserir a função PWM podemos utilizar a seguinte configuração:

PWM(INP, PW_0, PARA)

onde:

- *INP* – identificação do valor a ser modulado, geralmente o sinal de controle variável armazenado em uma *word*;
- *PW_0* – identificação da saída digital onde o sinal PWM estará disponível para controlar dispositivos externos;
- *PARA* – identificação dos endereços dos parâmetros do controlador - 5 *words* sendo o primeiro deles o *T_MOD* e o restante utilizado internamente (exemplo: %MW20 a %MW24). O parâmetro *T_MOD* é utilizado para definir o período de modulação (inverso da frequência do PWM). Este parâmetro é expresso em 1/100 s podendo ser ajustado entre 0 e 32767. *T_MOD* deve ser maior que o tempo necessário

para executar um ciclo de varredura sendo ajustado ... e pode ser ajustado...

A Figura 7.17 apresenta um exemplo de aplicação, sendo que nesse caso foi utilizada a configuração do PWM como:

PWM(%IW0.2, %QW2.0, %MW20:24)

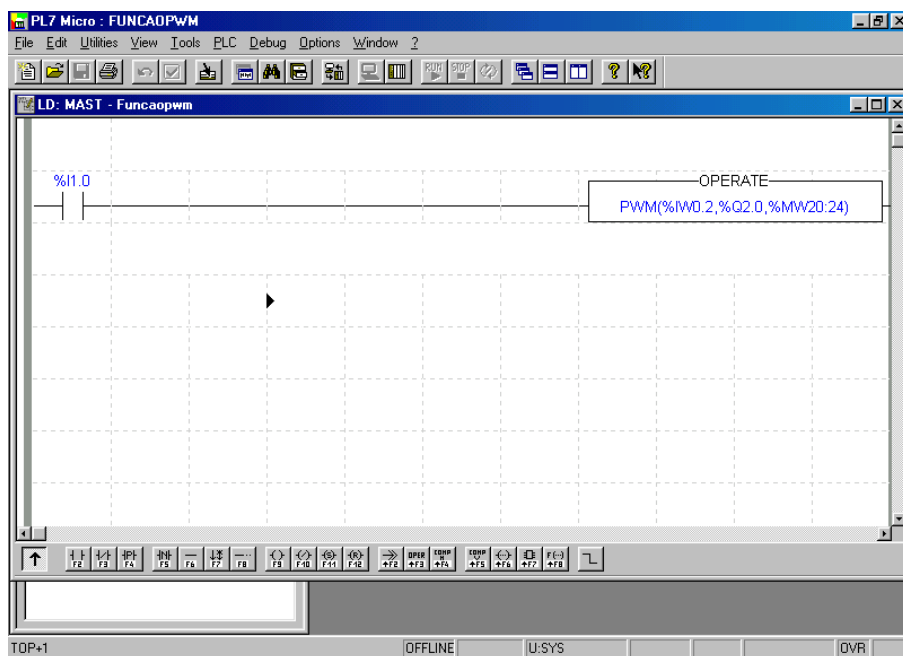


Figura 7.17 – Exemplo de Programação da Função PWM (CLP Modicom – Telemecanique®)

7.6 – Funções Especiais no PLC OMRON.

A programação do CLP OMRON usando o software CX-One permite implementar o controle PID (Proporcional, Integral e Derivativo). Para isso é necessário inicialmente configurar o módulo de entrada e saída analógica, o qual será detalhado a seguir.

Entradas e Saídas Analógicas

O módulo CJ1W-MAD42 (Figura 7.18) possui 4 entradas analógicas e 2 saídas analógicas. O endereço de acesso depende da configuração manual realizada no módulo (*switches*).



Figura 7.18 – Módulo de Entrada e Saída MAD42

As entradas e saídas são variáveis de 16 bits (padrão UINT), podendo apresentar valores de 0a 65535 (formato decimal), 1 a 9999 (formato BCD) ou ainda 0001 a 270F (formato hexadecimal).

Os ranges e valores de medição estão apresentados no Manual (“Guia Rápida Del Módulo Analógico CJ1W-MAD42”) nas páginas 4 a 7.

Para que as entradas e saídas analógicas funcionem é obrigatório habilitar as entradas e saídas clicando no IO Table e depois no módulo MAD42. É necessário também definir as faixas de tensão (por exemplo, 1 a 5 V ou 4 a 20 mA).

No caso do modelo do laboratório a identificação do endereço do módulo é ajustada por meio de chaves, sendo que se os switches foram ajustados em $S_1 = 2$ e $S_2 = 1$ resulta em endereço = $2000 + S_1 \cdot 10^2 + S_2 \cdot 10^1 = 2210$. Assim, nesse exemplo, as entradas do módulo poderão ser acessadas nos endereços 2215 a 2219 enquanto as saídas nos endereços 2201 e 2202.

Um exemplo de lógica para transferir dados de entrada e saída analógica é representado na Figura 7.19. No exemplo, um comando de entrada qualquer (P_On, por exemplo) permite habilitar a função MOV por meio da qual é realizada a leitura da entrada no endereço 2005, transferindo o valor binário correspondente para a saída analógica do endereço 2001. Na operação MOV podem ser utilizadas memórias (exemplo: D300, D400), valores constantes (por exemplo, #7FFF, caso queira transferir um valor hexadecimal para uma específica saída analógica), entre outros.

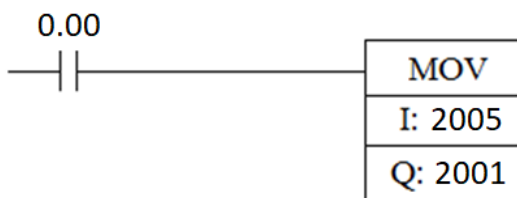


Figura 7.19 – Exemplo de Operação MOV para Leitura em Entrada e Escrita em Saída

Para habilitar o módulo MAD42 é necessário setar o bit menos significativo do endereço do módulo. Um exemplo é ilustrado na Figura 7.20



Figura 7.20 – Exemplo de Habilitação do Módulo de Entrada e Saída Analógica Setando o Bit Menos Significativo

Uma forma de testar o funcionamento da lógica desenvolvida pelos exemplos das Figura 7.19 e 7.20 consiste em conectar uma fonte de alimentação externa num determinado endereço de entrada e monitorar a saída escolhida. Um exemplo, é ilustrado na Figura 7.21, nesse caso usando endereço de entrada 2005 e endereço de saída 2001.



Figura 7.21 – Testando Módulo de Entrada e Saída Analógica

Variando a amplitude de entrada de 0 a 10 V (cuidado para não exceder a tensão de 10 V para evitar queimar o módulo de I/O Analógico) é possível monitorar os valores obtidos na entrada e saída do PLC, bem como a tensão produzida na saída. Assim é possível determinar a faixa de

valores de entrada e saída (range) e os correspondentes valores hexadecimais produzidos, podendo com isso ser determinada a resolução dos conversores A/D e D/A. Pode ser usada uma fonte externa ou ainda a própria fonte do painel, ajustada por meio de um potenciômetro.

Controle PID no PLC OMRON

O algoritmo clássico do controle PID é tipicamente realizado como ilustrado no diagrama de blocos da Figura 7.22.

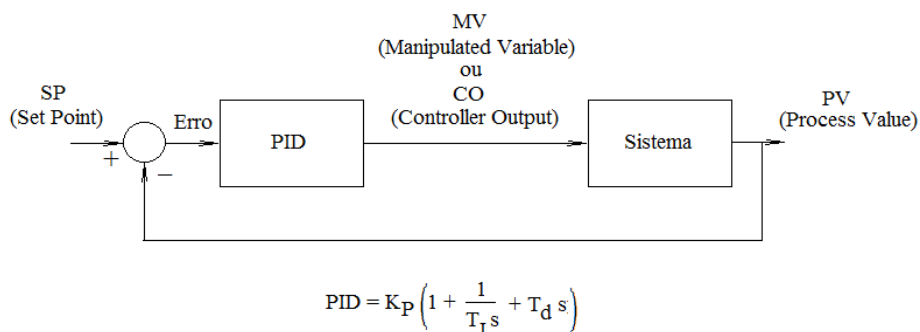


Figura 7.22 – Controle PID Clássico

O algoritmo de controle PID no PLC OMRON possui alguns diferenciais e pode ser implementado conforme o diagrama de blocos da Figura 7.23.

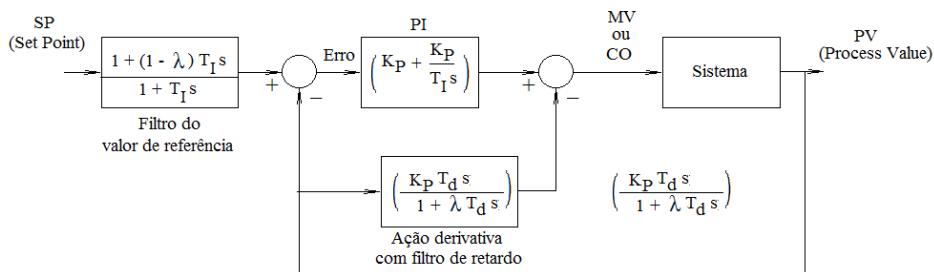


Figura 7.23 – Controle PID no PLC OMRON

Observa-se que:

- a parte derivativa é implementada separadamente e considera a inclusão de um filtro de retardo; essa característica permite que o controle PID minimize as ações bruscas na parte derivativa, bem como permite que essa componente atue apenas na componente de saída do sistema (PV – *Process Value*);

- um filtro passa-baixa é adicionado no *set point* (SP) permitindo que a variável de entrada modifique-se de forma mais suave, minimizando oscilações após mudanças bruscas no sinal de referência (SP).

O parâmetro λ permite ajustar o atraso na componente do filtro do valor de referência bem como no filtro de retardo da ação derivativa.

A implementação do controle PID no software CX-One pode ser realizada utilizando o exemplo ilustrado na Figura 7.24. Note que no bloco do PID devem ser definidos os endereços do sinal a ser controlado (I = *Process Value*), do primeiro endereço de configuração do PID (por exemplo, o endereço de memória D300) e do endereço da saída de controle (O = saída de controle – *Manipulated Variable*).

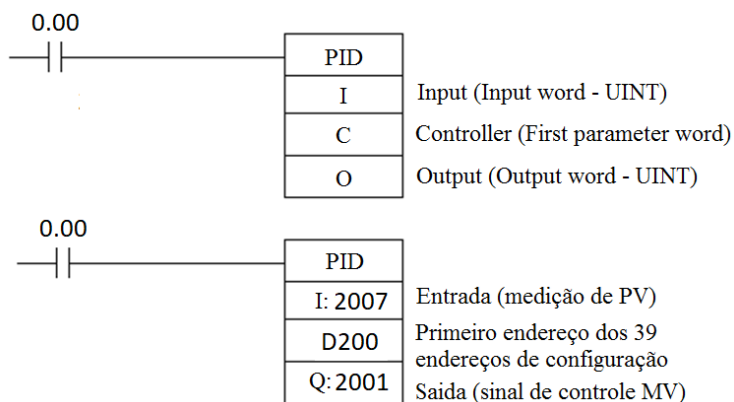


Figura 7.24 – Programando o Controle PID no Software CX-One

A configuração dos parâmetros do PID bem como o endereço do valor do *Set Point* (referência de entrada) são definidos a partir do endereço especificado em C (no exemplo, um endereço de memória D300). Assim, se o valor de referência é definido por um valor de entrada externo (exemplo, uma tensão controlada externamente) seu valor deve ser lido e atualizado continuamente no endereço D300 utilizando a função MOV.

Para carregar os valores iniciais dos parâmetros do PID (KP, TI, TD), do tempo de amostragem (exemplo: TS = 0,01 s) e dos filtros do PID (parâmetro λ), deve-se também utilizar a função MOV, para carregar os valores constantes nos endereços desejados, conforme a tabela de referência Tabela 7.3. Outra opção é carregar os parâmetros em tempo real usando o atalho ALT 3 e entrar com os valores na janela, para os respectivos endereços.

Endereço ou Posição de Memória	Parâmetro	Função
%Ci	SV	Entrada de referência (setpoint)
%C(i+1)	KP	Ganho proporcional (#0 a #65535) ou (#1 a #9999)
%C(i+2)	TI	Constante de tempo de integração em unidade 1/100 s; valores entre #1 e #8191. Integração é zerada para #9999
%C(i+3)	TD	Constante de tempo derivativa em 1/100 s; valores entre #1 e #8191. Derivativa é zerada para 0.
%C(i+4)	TS	Tempo de amostragem em 1/100 s; valores #1 a #9999 (0.01 a 99.99 s)
%C(i+5) bits 4 a 15	λ	2-PID Parameters- constantes dos filtros
%C(i+5) - bit 3	MV designation	0 - saída é 0% quando TI = #9999 (sem integração); 1 - saída é 50% quando TI = #9999 (sem integração)
%C(i+5) - bit 1	PID parameters atualization	Define o instante em que os parâmetros do PID são atualizados: 0 - no início da execução do PID 1 - em cada tempo de amostragem
%C(i+5) - bit 0	PID forward / reverse	Determina a direção da ação proporcional: 0 - Reverse action 1 - Forward action

Tabela 7.3 – Configuração dos Parâmetros do Controlador PID – 1ª Parte (CLP OMRON®)

Endereço ou Posição de Memória	Parâmetro	Função
%C(i+6) bit 13 e 14	Integral Manipulated Variable	Determina o estado inicial da componente MV quando o PID é iniciado
%C(i+6) - bit 12	Habilita limite para controle	0 - Desabilita 1 - Habilita
%C(i+6) Bit 8 a 11	Range da entrada	0: 8 bits; 1: 9 bits; 2: 10 bits;... 8: 16 bits
%C(i+6) Bit 4 a 7	Integral and Derivative Unit	1: múltiplo do tempo de amostragem; 0: unidade 100 ms
%C(i+6) Bit 0 a 3	Range de Saída	0: 8 bits; 1: 9 bits; 2: 10 bits;... 8: 16 bits
%C(i+7)	MV lower limit	0000 a FFFF
%C(i+8)	MV upper limit	0000 a FFFF
%C(i+9) a C(i+38)	Reservado	Words reservadas para cálculos do PID

Tabela 7.4 – Configuração dos Parâmetros do Controlador PID – 2ª Parte
(CLP OMRON®)

7.7 – Exercícios de Aplicações.

Nessa seção serão propostos alguns projetos visando aplicar os conceitos apresentados relacionados à utilização de funções especiais. Adicionalmente nesse projeto espera-se que do aluno um interesse pela realização de pesquisa bem como consulta ao manual de programação do PLC OMRON, visando conhecer funções especiais disponíveis para o modelo, o qual será utilizado nas aplicações em laboratório.

Exercício 1 – *Bargraph* – Uma Aplicação de Entrada Analógica

Nessa aplicação deseja-se desenvolver um sistema que monitore uma entrada analógica do CLP. Para isso, será necessário entender como identificar entradas analógicas e monitorá-las por meio da programação.

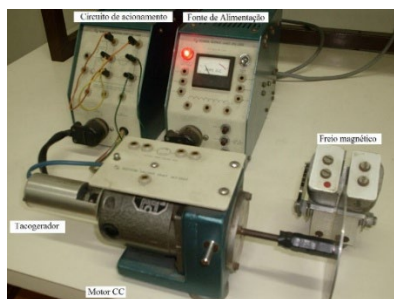
Deseja-se adicionalmente que, a partir do sinal analógico lido, sejam controlados um conjunto de saídas digitais, conectadas em 4 a 8 lâmpadas de saída, formando um *bargraph*, ou seja, um gráfico de barras, onde a representação binária produzida nas lâmpadas seja dependente da amplitude do sinal analógico monitorado. Assim, quanto maior o sinal analógico medido, maior a quantidade de bits em nível lógico “1” produzidos nas saídas digitais.



Figura 7.25 – Bargraph com 8 lâmpadas

Exercício 2 – Controle PID de Velocidade

Elaborar um programa para implementar o controle PID de velocidade de um motor ou servomotor de corrente contínua, tais como os motores ilustrados nas Figuras 7.26 (a) e (b).



(a)



(b)

Figura 7.26 – Sistemas para Controle de Velocidade

(a) Motor de Corrente Contínua

(b) Servomotor de Corrente Contínua

Para realizar o controle serão monitorados sensores de velocidade (tacogeradores) dos motores, conectando suas medições numa entrada analógica escolhida. Uma entrada analógica adicional, será escolhida para definir a entrada de uma tensão externa, a qual será correspondente ao valor do *Set Point* do sistema. O sinal de controle produzido pelo PLC será enviado para uma saída analógica, por meio da qual será possível controlar a velocidade do motor. Para isso, utilizar-se-á um circuito amplificador transistorizado o qual irá aumentar o nível de corrente, mas mantendo praticamente a mesma tensão (exceto por uma eventual perda de tensão no circuito eletrônico).

O projeto consiste em implementar o controle PID e sintonizar os parâmetros KP, TI, TD, TS e λ para garantir um funcionamento adequado, fazendo o sistema responder de forma rápida e com reduzida oscilação, além de não apresentar erro estacionário (ou erro estacionário relativamente reduzido).

Qualquer estratégia pode ser adotada para sintonia do PID.

A título de referência, propomos a seguir uma técnica, baseada no método de Ziegler-Nichols, mas que permite implementar o controle por tentativa e erro. A estratégia consiste nos seguintes passos:

- eliminar as ações integral e derivativa, colocando T_d e $K_i = 1/T_i$ em seu mínimo valor;
- ajustar o ganho proporcional K_p em um baixo valor e colocar o controlador em automático;
- aumentar o ganho K_p até que oscilações contínuas ocorram após uma pequena mudança no valor desejado ou na carga;
- reduzir K_p à metade;
- diminuir T_i lentamente até que as oscilações ocorram novamente; ajuste T_i igual a três vezes esse valor;
- aumentar T_d até que oscilações contínuas ocorram; ajustar T_d em um terço desse valor.

Qualquer estratégia pode ser adotada para sintonia do PID.

Na implementação do controle, sugerem as seguintes etapas:

- a) ensaio em malha aberta com o motor, como ilustrado na Figura 7.27 ou Figura 7.28. Note que utiliza-se uma fonte externa para produzir o *Set Point* e utilizam-se multímetros para verificar as medições do sinal de controle enviado para o motor, bem como do sinal do sensor de velocidade (tacogerador).
- b) Elabore um programa para que o valor da entrada I:2005 (fonte de alimentação) seja transferido simultaneamente para a saída Q:2001 e para uma memória interna D300. Transfira também a medição do sinal do tacogerador da entrada I: 2007 ou I:2006 para uma memória, por exemplo D500 (função MOV).

- c) Garanta a limitação no valor medido na saída do tacogerador em 10 V. Para isso, limite a tensão de saída aplicada no motor. Sugestão: utilize uma função de comparação. Quando o valor da saída for maior que 10 V, utilize função MOV para fixa-la no máximo valor.

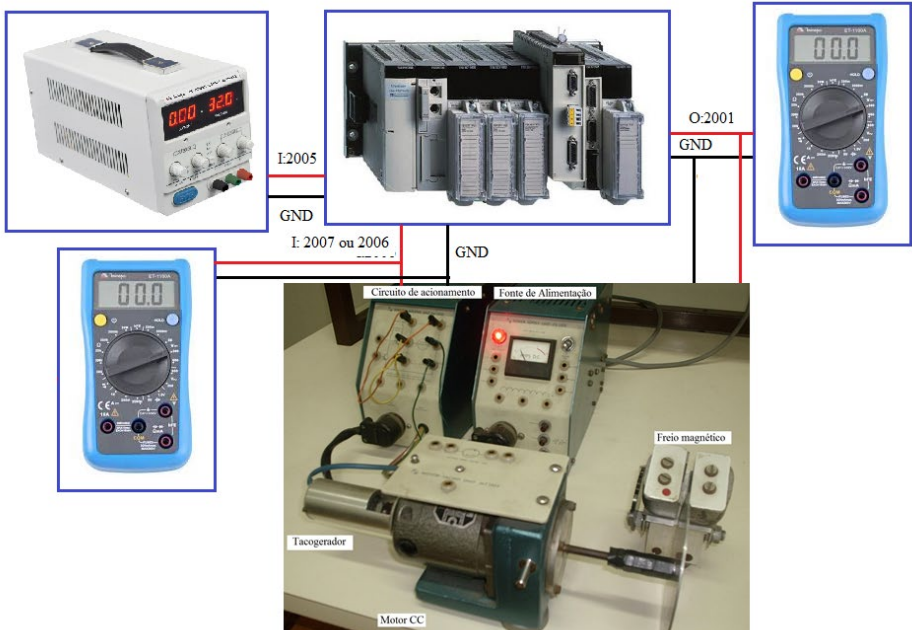


Figura 7.27 – Ensaio em Malha Aberta no Motor CC

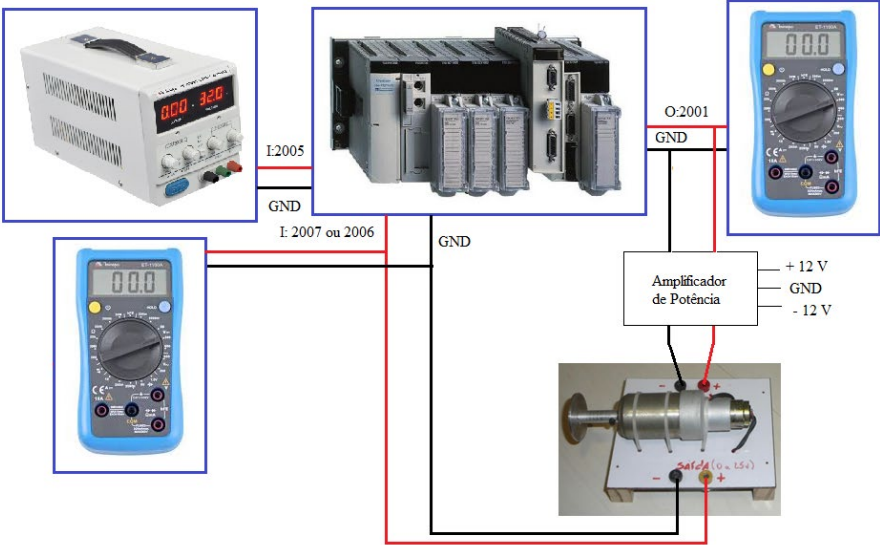


Figura 7.28 – Ensaio em Malha Aberta no Servomotor CC

- d) Elabore um programa utilizando controle PID para o sistema de controle de velocidade conforme ilustrado na Figura 7.29.

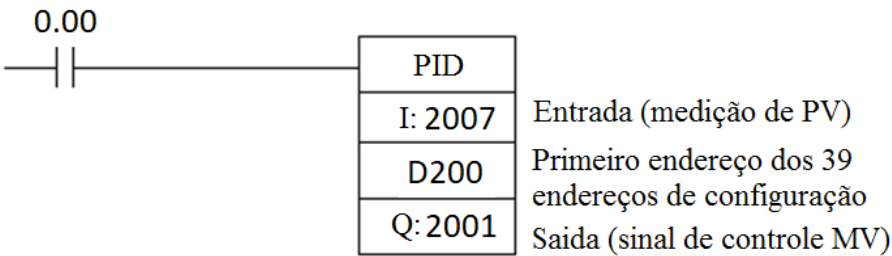


Figura 7.29 – Programando o Controle PID

No PID:

- o *Set Point* é ajustado por meio de fonte de alimentação conectada na entrada I:2005 e transferida por função MOV para a memória D300;
- ajuste os valores iniciais para os ganhos proporcional (KP), constante de tempo de integração (TI) e constante de tempo de integração (TD) respectivamente para os valores 000A, 1FFF (máximo valor) e 0000; para isso utilize funções MOV e considere que esses parâmetros são ajustados diretamente pelos endereços D301, D302 e D303 conforme a tabela de configuração do PID;
- ajuste também o tempo de amostragem em $T_s = 1$;
- analise o funcionamento do controlador PID e faça a sintonia dos parâmetros para otimizar o controle;
- acompanhe os parâmetros do PID monitorando os endereços específicos; para isso digite ALT 3 e entre com os endereços específicos D300 a D308.

8 – PROGRAMAÇÃO EM SFC / GRAFCET.

Neste capítulo serão apresentados os aspectos de programação de CLPs utilizando linguagem Grafcet (Gráficos Funcionais de Comandos Etapa Transição) ou SFC (*Sequential Function Chart*), ou seja, Sequenciamento Gráfico de Funções.

8.1 – Grafcet e SFC.

O Grafcet tem origem na França em meados dos anos 70. Derivado das redes de Petri, torna-se mais viável para modelagem de sistemas automatizados devido à facilidade de interpretação que apresenta.

Apesar do SFC ser uma forma simplificada do Grafcet, não existem grandes diferenças no modelamento utilizando as duas linguagens. Uma diferença relevante é que o SFC permite apenas um passo inicial por rede, enquanto o Grafcet permite mais de um.

As linguagens de programação em Grafcet ou SFC são preferencialmente utilizadas na forma gráfica, devido à facilidade de uso. Entretanto, esta é a única linguagem gráfica da norma que também pode ser programada de forma textual.

O Grafcet é uma simbologia gráfica com arcos orientados que permite implementar de forma eficiente algoritmos para controladores lógicos programáveis, incorporando todas as funções disponíveis para o Ladder, mas permitindo obter um grafo simples e com representação muito clara.

No Grafcet os arcos orientados interligam **etapas e transições**. As etapas, representadas por quadriláteros, são estados no qual o comportamento do circuito de comando não se altera frente a suas entradas e saídas. Transição representa a possibilidade de evolução do Grafcet de uma situação para outra, ou seja, de uma etapa para outra. A

passagem de uma situação à seguinte só é possível com a validade de uma transição.

A etapa que se torna ativa logo após o início de funcionamento do sistema é chamado de etapa inicial (duplo quadrilátero). A execução dos comandos em cada etapa é definida através de **ações** e as regras de evolução sequencial são definidas por **receptividades**. As **ações** e **receptividades** definem formalmente o comportamento dinâmico dos elementos comandados. As ações representam os efeitos que devem ser obtidos sobre o mecanismo controlado em uma determinada situação. A receptividade é a função lógica combinacional associada a cada transição.

A sequencialização do Grafcet é definida por arcos orientados, sempre no sentido de cima para baixo (em casos diferentes deste, recomenda-se utilizar flechas para orientação).

A Figura 8.1 apresenta um exemplo de aplicação em Grafcet aplicada a um sistema de controle de batelada.

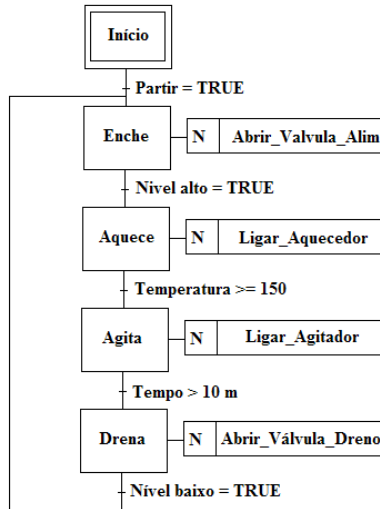


Figura 8.1 – Um exemplo de aplicação programada por Grafcet

8.2 – Ações.

As ações nos programas em Grafcet são representadas por um retângulo contendo três campos (Figura 8.2): Qualificador, Ação e Variável de Indicação. O qualificador determina quando a ação será executada. Uma ação pode ser uma simples variável booleana ou um conjunto de instruções a serem processadas.

Cada passo pode ter várias ações ou nenhuma. Um passo sem ação associada pode ser usado simplesmente para aguardar até que a condição seguinte seja satisfeita, funcionando como uma espera.

Ações complexas podem ser definidas fora do Grafcet, usando as linguagens LD (*Ladder Diagram*), FBD (*Function Block Diagram*), IL (*Instruction List*), ST (*Structured Text*) e o próprio SFC.

A variável de indicação é opcional e serve para mostrar o nome de uma variável que é modificada dentro do corpo da ação, com o propósito de indicar que uma ação foi finalizada, uma saída foi acionada, ocorreu um *time-out* ou um erro.

Qualificador	Ação	Variável de Indicação
P	Abrir válvula	OK

Figura 8.2 – Formato e exemplo de ação

Tipos de Ação:

- Ordem Contínua Sem Retenção (Figura 8.3) - Ordem de comando não retentiva cuja emissão depende apenas da ativação da etapa à qual estiver associada. Caso nenhum qualificador seja especificado, entende-se como ordem contínua sem retenção (default).

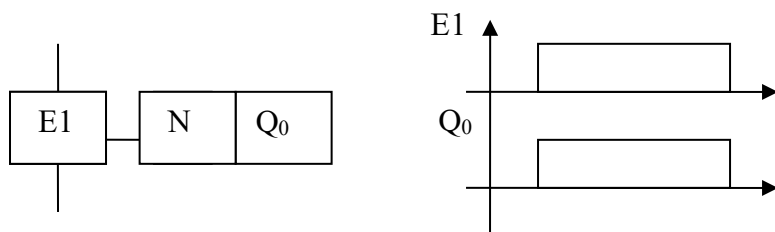


Figura 8.3 – Ordem contínua sem retenção

- Ordem Contínua Com Retenção (S ou R) (Figura 8.4) - Ordem de comando não retentiva permitindo ação de setar (S) ou resetar (R). Esta ação se mantém nas etapas consecutivas, exceto se outra ação com retenção oposta ocorrer.

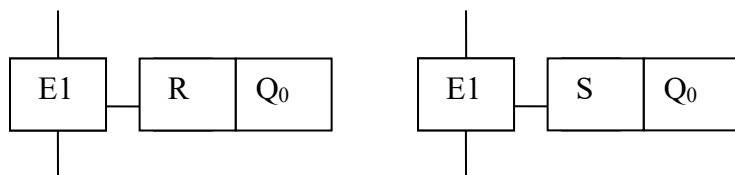


Figura 8.4 – Ordem contínua com retenção (S e R)

- Ordem Condicional (Figura 8.5) - A execução depende da ativação da etapa à qual estiver associada, depende também de uma outra condição lógica que a satisfaça. A condição X pode ser o estado de uma entrada, uma saída ou um bit de memória.

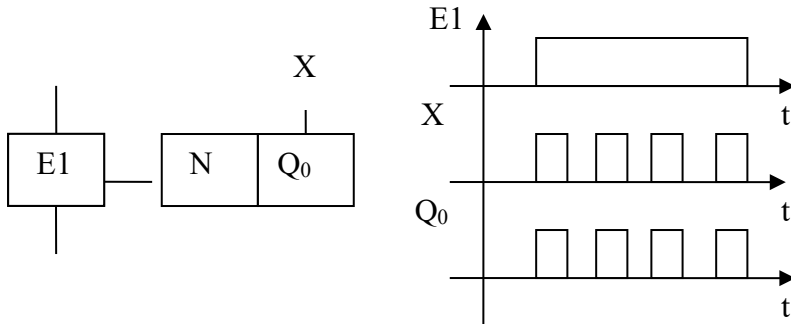


Figura 8.5 – Ordem condicional

- Ordem com retardo (D) (Figura 8.6) - Ordem condicional em que a dependência é associada a um retardo de tempo. Similar ao TON.

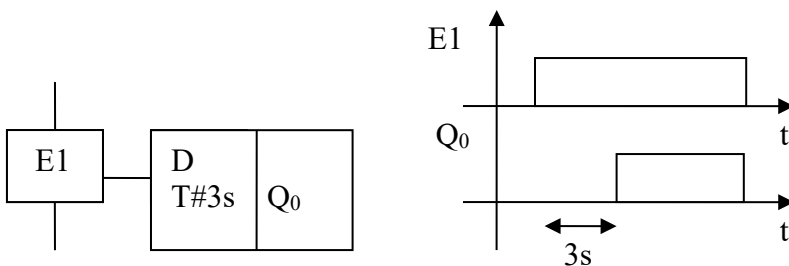


Figura 8.6 – Ordem com retardo

- Ordem limitada no tempo (L) (Figura 8.7) - Ordem emitida logo após a ativação da etapa, porém sua duração é limitada a um valor de tempo específico. Similar ao TP.

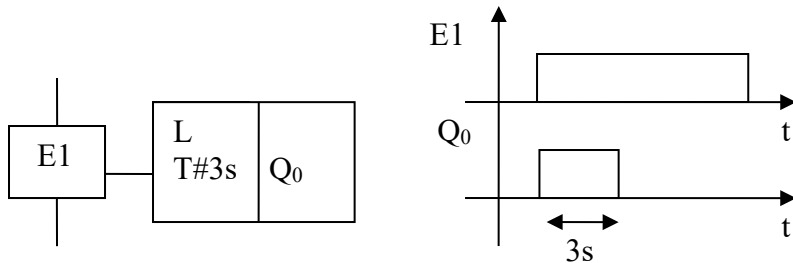


Figura 8.7 – Ordem limitada no tempo

- Ordem impulsional (P) (Figura 8.8) - Ordem similar à do tipo limitada no tempo, porém com tempo de duração infinitesimalmente pequena (na prática, ciclo de varredura).

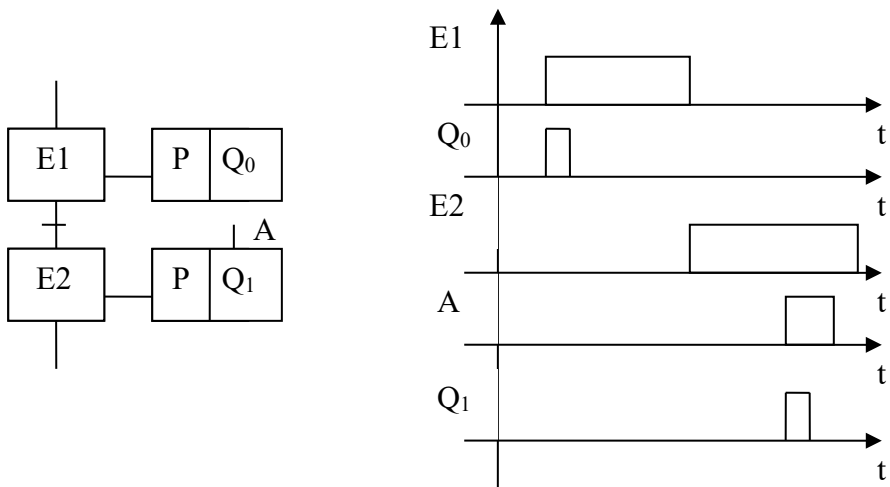


Figura 8.8 – Ordem impulsional

→ Em Diversas Etapas (Figura 8.9) - Repetição da mesma ação em mais de uma etapa. Note também que é possível executar mais de uma ação numa mesma etapa.

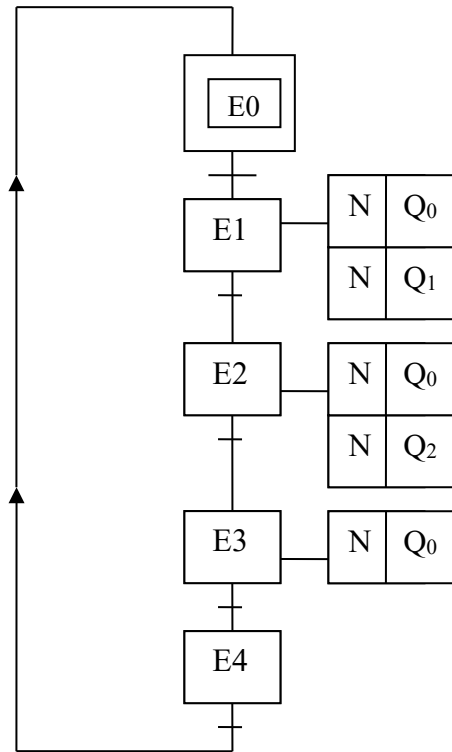


Figura 8.9 – Exemplo de aplicação usando ações repetidas em etapas consecutivas

Resultados similares para a saída Q_0 ilustrados na Figura 8.9 podem ser obtidos utilizando ações retentivas como na Figura 8.10.

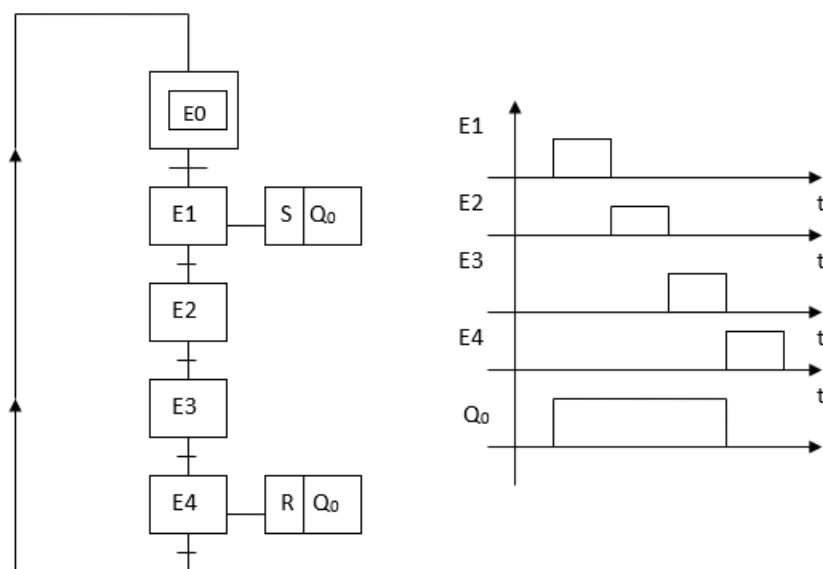


Figura 8.10 – Exemplo de aplicação usando ações retentivas

→ Etapas simultâneas (AND) (Figura 8.11) – Permite a execução de ações simultâneas desenvolvidas em etapas separadas.

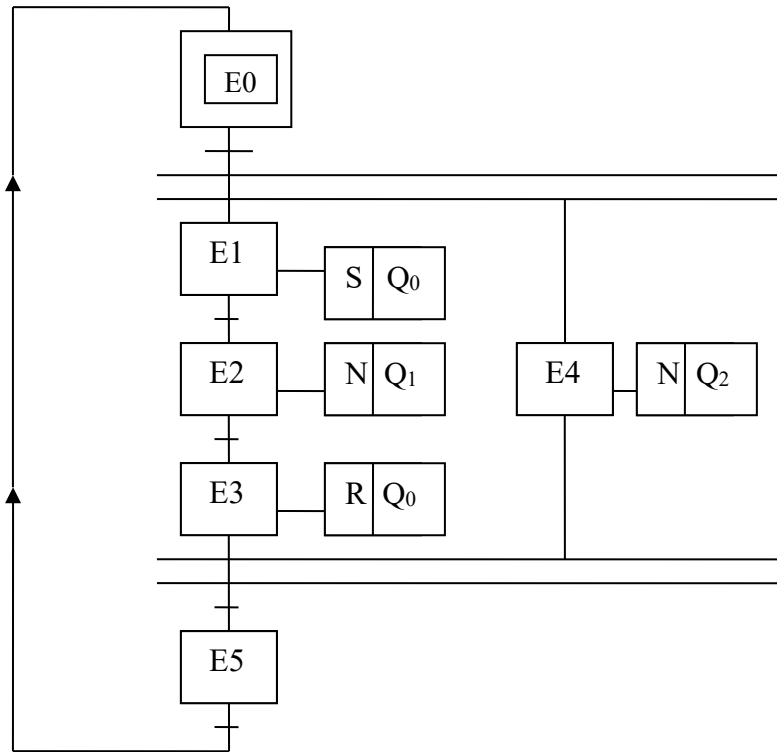


Figura 8.11 – Exemplo de aplicação usando etapas simultâneas

- Etapas separadas (OR) (Figura 8.12) – Permite a execução de ações selecionadas por transições a partir de um mesmo ponto. A etapa que será efetuada será aquela cuja condição de transição ocorrer primeiramente, ou obedecer a uma condição de prioridade.

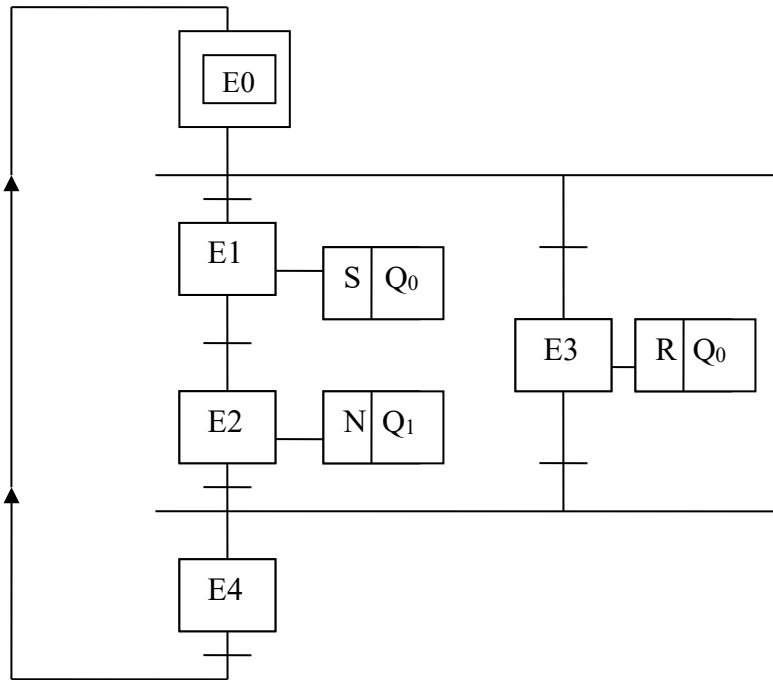


Figura 8.12 – Exemplo de aplicação usando etapas separadas (OR)

8.3 – Receptividades.

As receptividades podem ser o estado de uma entrada, uma temporização (T1, T2) ou a informação do estado de uma etapa. No exemplo da Figura 8.13 utiliza-se o paralelismo que permite que mais de uma etapa seja executada simultaneamente.

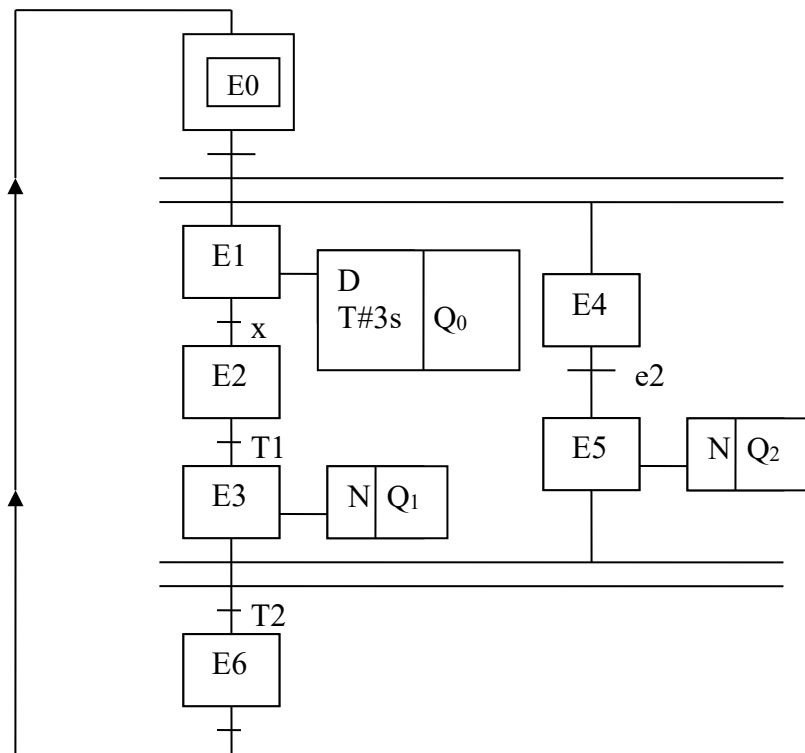


Figura 8.13 – Exemplo de utilização de vários tipos de receptividades na transição entre etapas

8.4– Programação em SFC no CLP Omron.

O SFC e o Grafcet são linguagens de programação previstas na norma IEC1131. Contudo, dependendo do fabricante do controlador lógico programável (Siemens, Telemecanique, Klocner Moeller, dentre

outros) alguns detalhes de programação são diferentes. Esforços estão sendo centrados pelas empresas no sentido de desenvolver uma interface de programação que satisfaça totalmente as condições impostas pela norma.

Para efetuar a programação em SFC o usuário deverá selecionar a opção SFC logo após abrir um novo programa no software CX-Programmer.

As Figuras 8.14 e 8.15 ilustram a criação de uma aplicação em SFC utilizando o software CX-Programmer.

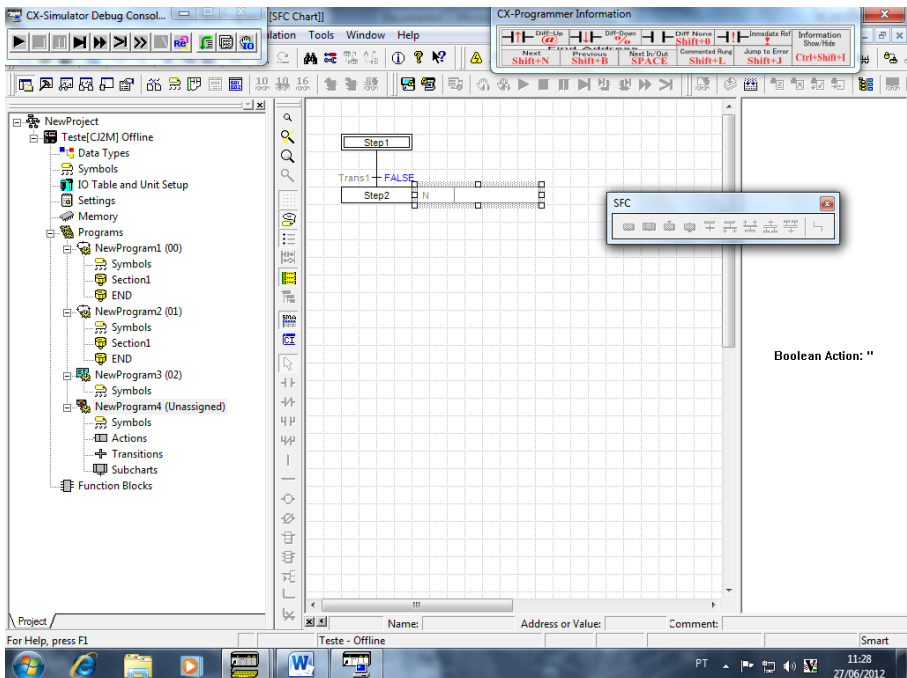


Figura 8.14 – Criação das primeiras etapas de um programa em SFC

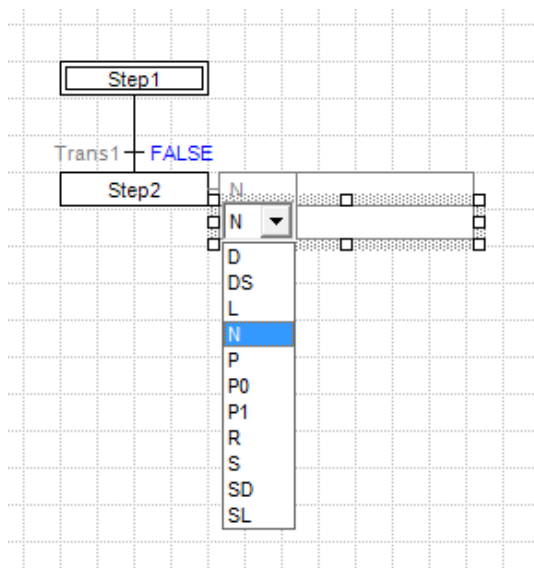


Figura 8.15 – Seleção de ação para uma etapa de um programa em SFC

8.5 – Exemplo de Aplicação.

Considere uma aplicação constituída de um sistema de controle de um robô com dois graus de liberdade conforme ilustrado na Figura 8.16.

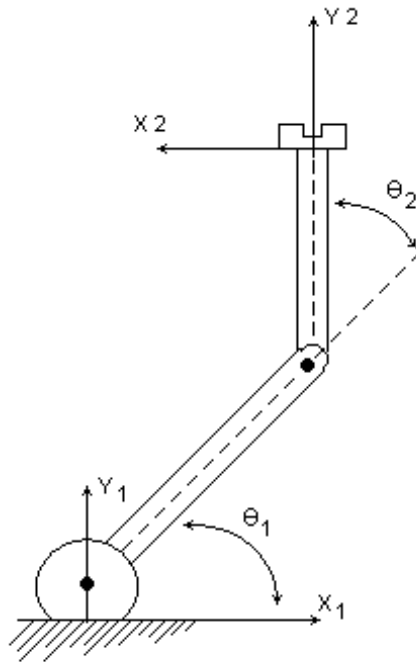


Figura 8.16 – Robô com dois graus de liberdade

O robô é constituído de uma garra (G) que permite manipular peças e dois braços mecânicos controlados por servomotores (M1 e M2) que podem ser acionados em ambos sentidos. O sensoriamento é efetuado através de sensores de posição (S1, S2 e S3). O comando de partida é efetuado pelo botão B1.

A lista de entradas e saídas do sistema é apresentada a seguir:

Entradas:

- B1 – Botão de partida.
- S1 – Sensor braço mecânico 1 na posição original.
- S2 – Sensor braço mecânico 2 na posição original.
- S3 – Sensor garra na posição de manipulação de peça.

Saídas:

- M1a - Servomotor M1 – Acionamento no sentido horário.
- M1b – Servomotor M1 – Acionamento no sentido anti-horário.
- M2a - Servomotor M2 – Acionamento no sentido horário.
- M2b – Servomotor M2 – Acionamento no sentido anti-horário.
- G – Acionamento da Garra

O controle do sistema é desenvolvido através de um programa sequencial que deverá ser desenvolvido em SFC de forma a efetuar a seguinte sequência de eventos:

- Etapa 0 - Acionamento do botão de partida foi efetuado setando memória M0 (ou M0 já setado anteriormente).
- Transição E0 → E1 – Condição: sensores S1 e S2 indicando que os braços manipuladores estão na posição original e M0 = 1.
- Etapa 1 – Acionamento do braço mecânico 1, através do servomotor M1 no sentido horário durante 10 segundos.
- Transição E1 → E2 – Condição: imediatamente após a conclusão da etapa 1. Isto significa que o braço mecânico 1 apresenta ângulo θ_1 desejado.

- Etapa 2 – Acionamento do braço mecânico 2, através do servomotor M2 no sentido horário durante 15 segundos.
- Transição E2 → E3 – Condição: imediatamente após a conclusão da etapa 2 e com sensor S3 acionado. Isto significa que o braço mecânico 1 apresenta ângulo θ_2 desejado e a garra está na posição adequada.
- Etapa 3 – Atraso de 8 segundos por segurança e em seguida acionamento da garra de manipulação de peças. Uma memória adicional pode ser utilizada no projeto.
- Transição E3 → E4,E5 – Condição: atraso de 8 segundos após a garra estar fechada (peça está pronta para ser transferida).
- Etapa 4 e 5 – Etapas simultâneas: acionamento dos motores M1 e M2 no sentido anti-horário com tempos de respectivamente 10 e 15 segundos.
- Transição E4 → E6 e E5 → E6 – Condição: sensores S1 e S2 indicando que os braços manipuladores estão na posição original.
- Etapa 6 – Abertura da garra ($G=0$) e liberação da peça.
- Transição E6 → E0 – Condição: Imediatamente (dependendo apenas do ciclo de varredura).

O diagrama em SFC construído para desenvolver esta aplicação está ilustrado a seguir na Figura 8.17.

Desenvolva o programa em SFC utilizando o software CX - Programmer.

Lembre-se que:

- O primeiro passo é definir as entradas e saídas apropriadas para o CLP Omron.
- Quando desejar-se que uma transição ocorra somente após a conclusão das ações definidas na etapa anterior, incluir condicionais na transição que garantam a conclusão das ações da etapa anterior antes da transição para a próxima etapa.
- Os tempos escolhidos são tipicamente maiores que os usuais (na prática) simplesmente para facilitar os testes.

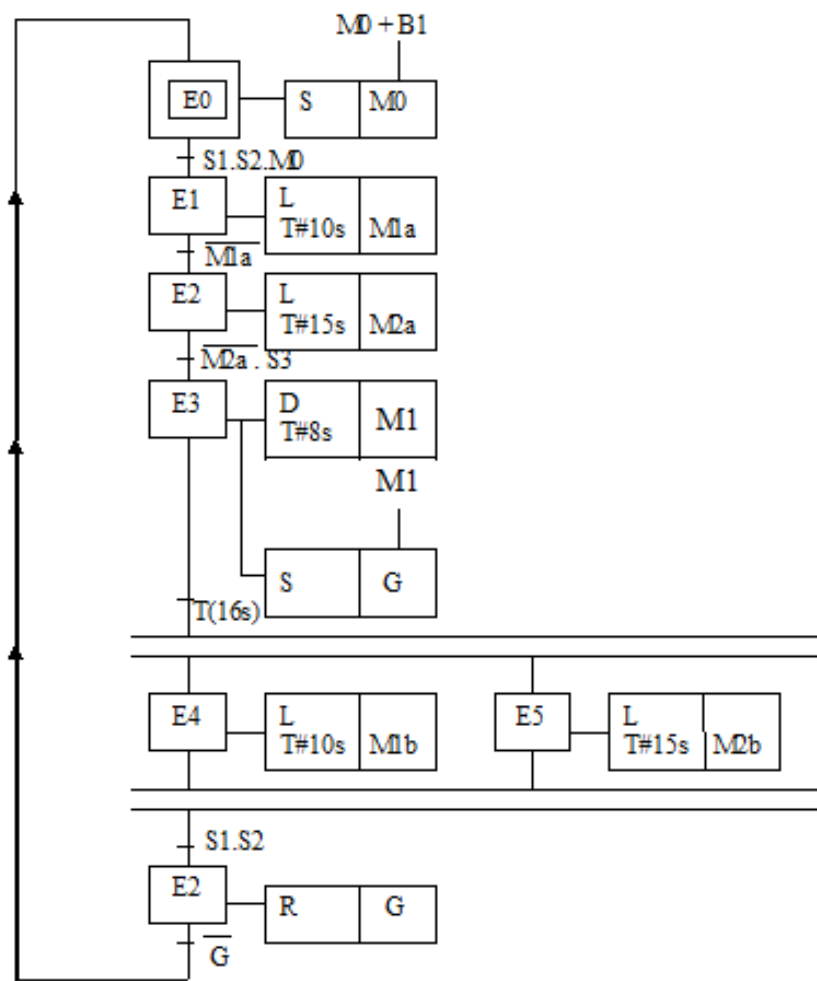


Figura 8.17 – Programa em Grafcet para desenvolvimento da aplicação

9 – REDES INDUSTRIAIS

9.1 – Redes Industriais para Automação Industrial.

A utilização de redes para conexão de equipamentos é cada vez mais presente na indústria. Particularmente na área de automação o uso de redes industriais proporciona várias vantagens, dentre as quais podemos citar: redução de custos com cabeamento e custos globais, simplicidade de instalação, operação e manutenção, menor número de módulos de entrada e saída é necessário e adicionalmente há maior capacidade de I/Os (número de pontos por porta no PLC), flexibilidade para expansão e introdução de novos módulos, facilidade de utilização de ferramentas para diagnóstico, variedade de soluções atualmente disponíveis e possibilidade de conectar dispositivos de diferentes fornecedores. Adicionalmente as redes industriais permitem atingir longas distâncias e a realização do controle físico distribuído. As principais dificuldades na instalação desses sistemas são o alto investimento inicial e a necessidade de adquirir conhecimento qualificado na área.

A Figura 9.1 ilustra a conexão de dispositivos num CLP (Controlador Lógico Programável) utilizando o sistema tradicional, ou ponto a ponto, onde cada ponto de entrada/saída (I/O) é conectado diretamente ao CLP (Figura 9.1 (a)) e a conexão utilizando uma rede industrial (Figura 9.1 (b)).

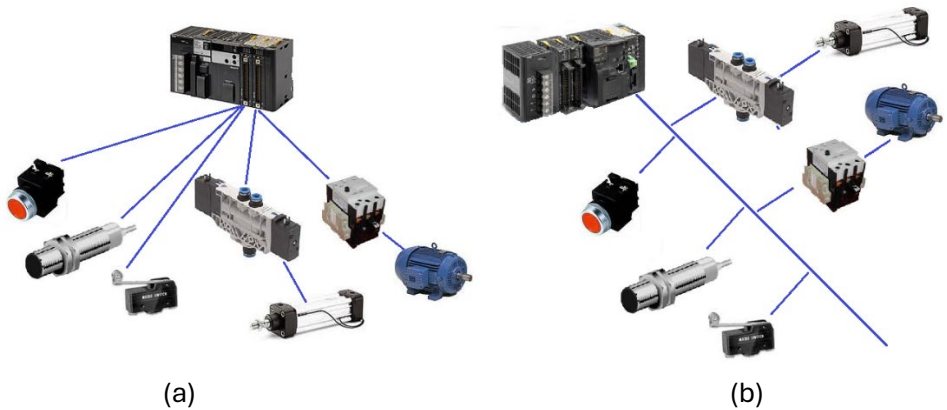


Figura 9.1 – Aplicação com CLP em rede e ponto a ponto.
(a) Configuração ponto a ponto (b) Configuração em rede

São requisitos importantes de redes industriais:

- manter a integridade e funcionalidade das operações e transferência de dados;
- permitir facilidade e segurança na aquisição de dados;
- permitir a interconexão de todos os dispositivos e softwares incluindo: módulos de aquisição de dados, CLPs, instrumentos, sistemas de supervisão (softwares SCADA - *Supervisory Control and Data Acquisition*), etc.;
- expansibilidade;
- compatibilidade entre produtos de vários fornecedores;
- disponibilidade de suporte técnico.

As redes industriais foram criadas originalmente para permitir a conexão dos controladores lógicos programáveis (CLPs) aos barramentos de chão de fábrica (*fieldbus*).

Alguns tipos de protocolos para redes industriais mais utilizados são:

- AS-I;
- PROFIBUS, que pode ser classificado em dois tipos distintos: PROFIBUS DP e PROFIBUS PA;

- Fieldbus Foundation (H1); Fieldbus Foundation (HSE); Fieldbus Foundation (H2);
- Modbus;
- CAN (*Controller Area Network*) e outros protocolos de alto nível baseados em rede CAN, como por exemplo as redes CAN intraveiculares (*In-Vehicle Networking*), destinadas para aplicações automotivas e a redes CANopen, DeviceNet e SDS;
- ETHERNET / IP;
- Outras (CONTROLNET, PROFINET, P-NET RS-485, P-NET RS-232, WORLDVIEW (MPS, MCS, SubMMS), INTERBUS, INTERBUS TCP/IP, INTERBUS SUBSET, SWIFTNET TRANSPORT, SWIFNET FULL STACK).

Algumas das principais características das redes AS-i, PROFIBUS, CAN, DeviceNet e ETHERNET são apresentados a seguir.

Redes AS-I

A rede AS-I (*Actuator Sensor-Interface*) é a de menor custo para ser instalada e uma das mais simples de ser operada e programada pelo usuário. Tem como principal característica a capacidade de trafegar informações de natureza discreta, ou seja, dados provenientes de sensores discretos e de atuadores ON/OFF.

Como a maioria das redes industriais disponíveis no mercado, o mestre faz a varredura das informações de entrada, executa o software de aplicação do usuário e atualiza as informações de saída.

As principais características da rede AS-I são:

- permite alimentação e comunicação trafegadas no mesmo par de fios, uma sobreposta a outra;
- permite derivações a qualquer momento, inclusive com a rede energizada;
- permite montagem em várias topologias como estrela, linear ou árvore;
- permite, no máximo, quatro bits de informações de I/O, podendo ser bidirecionais (quatro entradas + quatro saídas);
- opera com taxa de transmissão fixa em 167,5 kbps.

A impedância da rede AS-I varia entre 70 Ω e 100 Ω para a taxa de transmissão de 167,5 kbps. O tempo de resposta máximo pode chegar a 5 ms (Versão 2.0), 10 ms (Versão 2.1) e 20 ms (Versão 3.0), caso a rede esteja com todos os escravos pontos de I/O sendo utilizados.

As principais limitações das redes AS-I são:

- as informações trafegadas na rede AS-I são limitadas a quatro bits de entrada e/ou quatro bits de saída, por escravo, que podem ser trafegados a cada ciclo de varredura;
- é estritamente uma rede mestre-escravo síncrona, com *polling* cíclico por escravo;
- máximo de 31 escravos (Versão 2.0), 62 escravos (Versão 2.1, com quatro entradas e três saídas por escravo) e 62 escravos (Versão 3.0, com quatro entradas e quatro saídas por escravo);
- comprimento limitado da rede de 100 m sem repetido, 300 m com repetidor ou 600 m com repetidor respectivamente para as Versões 2.0, 2.1 e 3.0.

Exatamente devido a essas limitações, a rede AS-I geralmente é interligada a outra rede de nível hierárquico superior para aumentar a quantidade e a velocidade das informações trafegadas no sistema. Assim, de acordo com as características da aplicação e do custo máximo a ser atingido, uma combinação gradual de diferentes sistemas de comunicação, tais como Ethernet, PROFIBUS e AS-I, oferece as condições ideais de redes abertas em processos industriais.

Redes PROFIBUS

O padrão PROFIBUS (PROcess FieldBUS) é um padrão aberto de rede de comunicação industrial utilizado em um amplo espectro de aplicações em sistemas contínuos e discretos (LUGLI *et al.*, 2010). As redes PROFIBUS são indicadas para realizar a comunicação entre dispositivos no nível de campo, ou seja, para conexão com módulos de E/S, transdutores, acionamentos (*drivers*), válvulas e painéis de operação.

As principais características das redes PROFIBUS são:

- total independência de fabricantes (padronização garantida pelas normas EN50170 e EN50524);

- transmissão em alta velocidade e comunicação eficiente em operações complexas;
- transmissão em meio físico usando padrões RS485, Manchester (similar à rede AS-I) ou fibra óptica, dependendo da aplicação;
- utiliza dispositivos mestres e escravos; os dispositivos mestres determinam a comunicação de informações no barramento, pode enviar mensagens sem uma requisição externa, sempre que possuir o direito de acesso ao barramento; os dispositivos escravos são remotos tais como módulos de I/O, válvulas, módulos para acionamentos de velocidade variável e transdutores; os escravos não têm direito de acesso ao barramento e só podem enviar mensagens ao mestre ou reconhecer mensagens recebidas quando solicitados.

O PROFIBUS é baseado em padrões reconhecidos internacionalmente, sendo sua arquitetura de protocolo orientada ao modelo de referência OSI (*Open System Interconnection*) conforme o padrão internacional ISO 7498.

Possui dois tipos de protocolos distintos: PROFIBUS DP e PROFIBUS PA.

PROFIBUS DP

O protocolo PROFIBUS DP - Periferia Descentralizada (*Decentralized Periphery*) é o mais frequentemente utilizado, sendo otimizado para conexão de baixo custo e alta velocidade no nível de sensor/atuador. Baseia-se na rede ModBus, sendo uma rede dedicada, desenvolvida pela Siemens. Permite comunicação com dispositivos de I/O discretos mas também a transmissão de sinais de 4 a 20 mA para aplicações de controle contínuo. Essa transferência é tipicamente cíclica com tempo de ciclo de leitura/escrita obrigatoriamente menor que o tempo de ciclo do controlador. O PROFIBUS DP necessita de aproximadamente 6 ms em 1,5 Mbits/s e menos que 2 ms em 12 Mbits/s para a transmissão de 512 bits de entrada/saída distribuídos em 32 estações.

O PROFIBUS DP é baseado na norma EM 50170 e define três tipos de dispositivos básicos:

- Mestre DP Classe 1 (DPM1): controlador central que troca informações com dispositivos de I/O distribuídos (escravos DP);
- Mestre DP Classe 2 (DPM2): são equipamentos de configuração, monitoração ou ferramentas de engenharia utilizadas para configurar a rede ou parametrizar escravos DP;
- Escravo DP: dispositivo periférico que interliga diretamente com os sinais de I/O ao mestre classe 1; os escravos podem ser módulos de entradas, saídas, *drivers*, válvulas, painéis de operação, entre outros.

Tipicamente utiliza meio de comunicação no padrão RS-485, em especial em sistemas discretos, ou fibra ótica para aplicações em sistemas que demandam grande imunidade a interferências eletromagnéticas e grandes distâncias.

PROFIBUS PA

O protocolo PROFIBUS PA - Automação de Processo (*Process Automation*) é um padrão menos frequentemente utilizado, sendo desenvolvido para utilização apenas em sistemas de transmissão de sinais de 4 a 20 mA ou HART para aplicações de controle de processos analógicos. Sua velocidade é fixa em 31,5 kbps e depende da versão DP para entrar em operação possuindo um custo maior se comparado à versão DP. Sua vantagem consiste na possibilidade de configuração e parametrização integradas no instrumento de campo e a padrões de supervisão avançados do mercado, como por exemplo FDT (*Field Device Tool*) e DTM (*Device Type Manager*).

Tipicamente utiliza meio de comunicação no padrão Manchester em especial para aplicações em sistemas de controle de processos.

Redes CAN

As redes CAN foram desenvolvidas inicialmente pela Bosch, em 1986, para aplicações automotivas. São redes de baixo custo, alta confiabilidade, relativa simplicidade e com características de tempo real adequado para aplicações em sistemas de automação industrial. Adicionalmente apresenta funcionalidades de priorização de mensagens

e apresenta protocolo de alto nível (HLP - *High Layer Protocol*) para as aplicações de acordo com as necessidades do usuário, pois seu padrão básico cobre somente as duas primeiras camadas do padrão ISO/OSI.

O protocolo tem base na técnica CSMA/CR (*Carrier Sense Multiple Access / Collision Resolution*) às vezes também chamado de CSMA/CD + AMP (*Carrier Sense Multiple Access / Collision Detection and Arbitrary on Message Priority*). Isso significa que, sempre que ocorrer uma colisão entre duas ou mais mensagens, a de mais alta prioridade terá o acesso ao meio físico assegurado e prosseguirá a transmissão.

As características básicas das redes CAN são as seguintes: 8 bytes de dados, velocidade de até 1 Mbit/s, priorização de mensagens, recepção *multicast* com sincronização, detecção de erros e sinalização e retransmissão automática de mensagens corrompidas.

A primeira versão (CAN 2.0A) especificava somente mensagens com identificadores padrão de 11 bits. A versão CAN 2.0B admite também mensagens estendidas com identificadores de 29 bits.

O protocolo CAN foi adotado em 1993/94 como padrão mundial ISO11898 pela *International Standardization Organization* (ISO).

Alguns dos protocolos derivados do CAN são:

- redes CAN voltadas para aplicações automotivas: Volcano, J1939 e CANKingdom;
- redes CAN voltadas para automação industrial: CANopen, *Smart Distributed Systems* (SDS), e DeviceNet.

Apesar de desenvolvidos por empresas e/ou grupos de empresas, os protocolos acima são abertos e permitem que qualquer um os utilize e desenvolva novos produtos sem ônus. Além disso, muitas ferramentas de software para programação, monitoração, configuração e módulos de hardware foram desenvolvidas para os protocolos citados e estão disponíveis comercialmente.

Redes DeviceNet

As redes DeviceNet seguem o modelo *Open System Interconnection* (OSI), um padrão da ISO (*International Standards Organization*) para redes de comunicação que são de natureza

hierárquica, as quais definem todas as funções necessárias da implementação física até o protocolo e metodologia para comunicar dados de controle e informação dentro e através de redes.

A rede especifica uma configuração com tronco, terminações e linhas secundárias e inclui até 64 nós lógicos selados ou abertos. A taxa de transferência é de 125 kbps, 250 kbps ou 500 kbps, selecionável dependendo do comprimento da rede.

O padrão DeviceNet firmou-se no mercado devido à diversidade de produtos oferecidos e também pela excelente solução técnica de uma rede produtor/consumidor, que poupa o meio físico das desgastantes trocas de dados inúteis. Juntamente com a rede Profibus DP, possui grande aplicabilidade no mercado nacional e internacional, pois possibilita a conexão de elementos discretos e sinais analógicos com alta taxa de transmissão e fácil configuração e programação de toda a rede.

A Tabela 9.1 mostra um comparativo dos principais parâmetros entre uma rede industrial DeviceNet e outras duas redes conceituadas no Brasil, AS-Interface e Profibus DP.

Redes ETHERNET/IP

A rede ETHERNET é a rede tipicamente utilizada na interligação de computadores, mas que também pode ser utilizada para interconectar dispositivos de automação industrial. A principal diferença das redes ETHERNET em relação às outras redes industriais é que, essas últimas são determinísticas, pois possuem tempos exatos para o tráfego da informação (tempos de varredura). Por outro lado, as redes ETHERNET são probabilísticas, ou seja, não possuem tempos exatos para o tráfego das informações (por exemplo, um computador conectado à Internet).

Característica	DeviceNet	Profibus DP	AS-Interface
Número de estações	64	125 (32 para trecho de rede)	31 (62 para AS ₂ 1 ou AS-I 3.0)
Comprimento	500m@125K	400m@500K	100m@5/10ms
Extensão	+ 500m/repet	+ 400m/repet	+ 100m/repet
Nº repetidores	3	4	2
Taxas	125, 250, 500 KB	configurável	5ms/10ms fixo
Telegrama de informações	64 bytes	244 bytes	8 bits
Comunicação	produtor / consumidor	mestre(s) / escravo	mestre(s) / escravos
Acesso assíncrono	sim, configurável para cada dispositivo	sim, configurável para cada dispositivo	não
Endereçamento	<i>dip</i> ou <i>software</i>	<i>dip</i>	<i>hand held/software</i>
Topologias	<i>line, branch line</i>	<i>line/stubs</i>	<i>line branch ring tree star</i>
Terminador	resistor	resistores ativos	sem
Meio físico	2 pares	1 par ou 2 pares	1 par
Blindagem	sim + dreno	sim + dreno	Sem
Alimentação	mesmo cabo	par separado ou mesmo cabo	mesmo par fios

Tabela 9.1 - Comparativo entre redes DeviceNet, Profibus DP e AS-Interface

Fonte: LUGLI, 2009.

O padrão TCP/IP surgiu no meio industrial após o ano 2000 e somente recentemente é que se tornou comercialmente utilizado em ambientes industriais.

Sua grande utilidade consiste na possibilidade de interligar todos os níveis da cadeia de suprimentos envolvendo um único e exclusivo padrão de rede, incluindo nível de gerência ou vendas até o chão de fábrica, e em tempo real, dando uma grande agilidade à produção e aumentando-a efetivamente.

Na arquitetura tradicional isso não é possível devido aos diferentes tipos de padrões de protocolos existentes no mercado, como DeviceNet, AS-i, Profibus DP, etc.

O protocolo de camada física utilizado pelo padrão TCP/IP é o IEEE 802.3 (Ethernet). Os primeiros meios de transmissão eram constituídos de pares metálicos grossos, cuja atenuação era muito grande. Depois surgiram os cabos coaxiais e em seguida os pares trançados. Hoje utilizam-se fibra óptica ou comunicação sem fio (*wireless*).

Houve também uma grande evolução nos conectores do hardware utilizado na rede. Atualmente utiliza-se o conector RJ45.

As taxas de transmissão aumentaram de 10 Mbps para valores da ordem de 100 Mbps e 10 Gbps. Contudo, a largura de banda não é suficiente para dar velocidade a determinadas aplicações.

9.2 – Redes Profibus DP.

Nesse experimento pretende-se demonstrar a configuração e montagem do CLP CJ2M para conexão de dispositivos utilizando redes DeviceNet e PROFIBUS DP.

Configurações Iniciais no Inversor de Frequência – Operação Manual e Automática

Nessa seção pretende-se demonstrar a configuração e montagem do CLP CJ2M para conexão de dispositivos utilizando o padrão de comunicação PROFIBUS DP e como meio físico um conector no padrão

RS-485. A aplicação consiste no acionamento de um motor de indução trifásico utilizando um inversor de frequência (Figura 9.2).

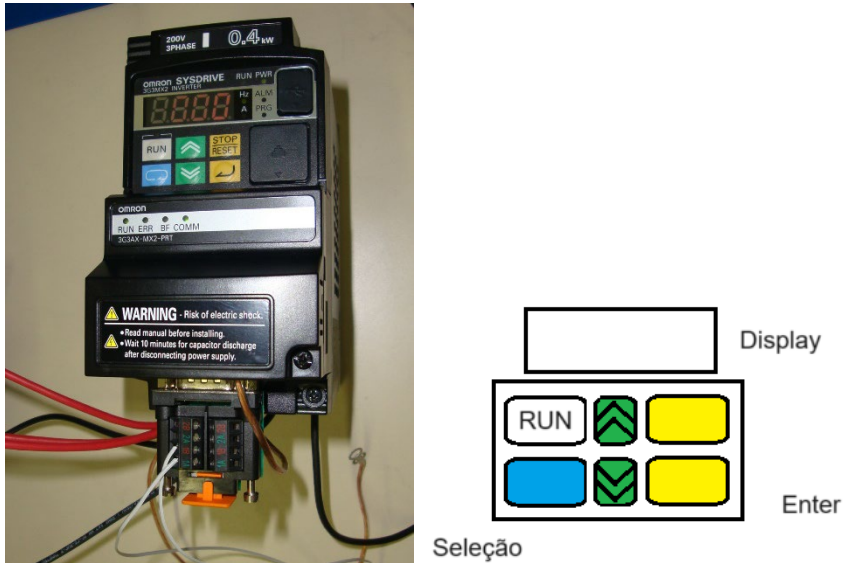


Figura 9.2 – Inversor de frequência e esboço de teclas de configuração

Contudo, antes de demonstrar o controle automático, pelo CLP, faremos a configuração para controle de velocidade manual do motor usando inversor de frequência.

A Figura 9.3 demonstra as conexões dos três enrolamentos do motor para trabalhar com a tensão nominal na conexão triângulo. A Figura 9.4 mostra a montagem do inversor na bancada do laboratório para acionar o motor de indução trifásico. A Figura 9.5 ilustra a conexão do inversor no CLP Omron.

Após realizar a montagem acione a chave Liga/Desliga. O LED da bancada acende.

Para realizar a operação manual do motor primeiro será necessário realizar as seguintes configurações:

- clicando no botão de seleção sequencialmente aparecerão os seguintes endereços de configuração:

- a) A0001 (usado para selecionar modo automático ou manual)
- b) b0001



Figura 9.3 – Conexões do Inversor de frequência acionando motor de indução trifásico

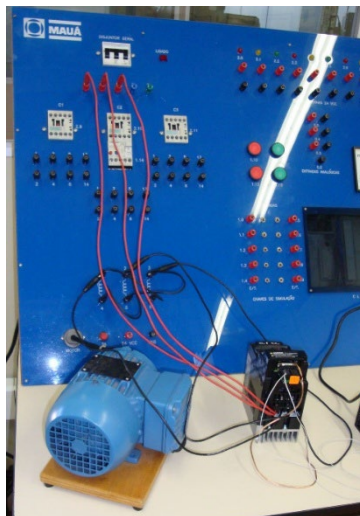


Figura 9.4 – Inversor de frequência acionando motor de indução trifásico

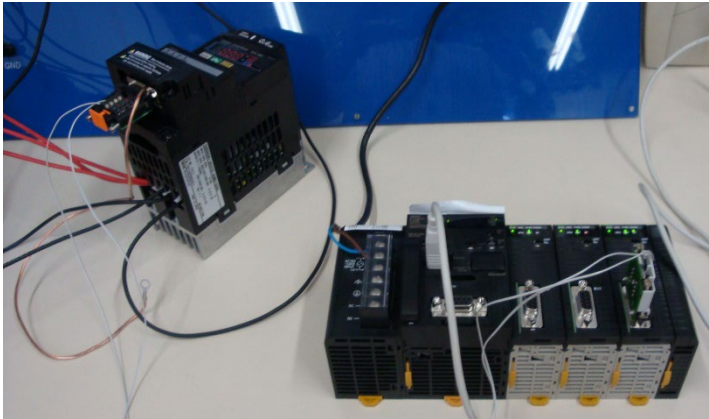


Figura 9.5 – Conexão de inversor de frequência em CLP Omron usando rede PROFIBUS DP

c) C0001

d) d0001 (configurações de monitoramento, permite monitorar a velocidade do motor associada à frequência)

e) f0001 (permite ajustar a frequência do inversor e com isso a velocidade do motor no modo manual)

Excursione acionando o botão de seleção até chegar em A0001. Clique em Enter e depois nas setinhas para ajustar A0001 = 02. Clicando nas setinhas vá para o endereço A0001 e ajuste A0002 = 02 também.

Essa seleção é necessária para o modo manual.

Agora excursione para o endereço f0001. Clique nas setinhas para ajustar a frequência (de 0 a 60). Clique em RUN e você irá observar a variação da velocidade. Ajustando o valor de 0 a 60 observa-se a variação na tensão eficaz do motor (0 a 220 V) e como consequência na velocidade do motor.

No próximo passo iremos deixar tudo preparado para realizar o controle do motor pela rede Profibus DP. Para isso precisam ser ajustados diversos bits de configuração no inversor de frequência.

Para isso faça os seguintes ajustes: B037 = 0 (para permitir que os diversos endereços de configuração estejam disponíveis), A0001 = 04, A0002 = 04 (esses dois bits permitem trabalhar no modo automático), c021 = 63, c022 = 63, c026 = 63, c028 = 16 (esses bits são importantes para configurar o endereço do módulo no PLC), p033 = 06, p180 = 2 e p182 = 1 (para trabalhar no modo *Basic Data*, como será explicado a seguir).

Em seguida iremos configurar a rede Profibus DP no computador.

Configurações para criação da rede PROFIBUS DP

A rede PROFIBUS DP é configurada por meio do software Configurator FDT, o qual pode ser acessado em Omron / CX-One / Configurator FDT. Na primeira vez em que o acesso ocorrer o software abre uma caixa de diálogo onde devemos realizar procedimento para carregar todos catálogos. Para isso, responda “Sim” ao procedimento de *Create Device Catalogue*.

O início da configuração é realizado selecionando *My Network* com o botão direito do mouse e *Add*. Selecione o módulo CJ1W - PRM21 – PROFIBUS DP V.1 Master. Verifique com atenção pois há nomes muito parecidos.

Selecione *Device Configuration* (ou clique diretamente no módulo) e clique em *Configure*. Se não aparecer nenhuma janela, pode ser que ela esteja sobreposta por outra janela. Ajuste PLC CJ2M e CPU31). Em *Start Address* mantenha a configuração original: 1. Em *Unit Number* selecione o endereço do módulo PROFIBUS no CLP, geralmente ajustado em *Unit Number* 4. Realize o *Test* (o módulo PRM21 é reconhecido) e *Save*.

Selecione *View / Device Catalogue*: abre-se uma janela na lateral direita. Selecione o device 3g3AX-MR12-PRT. Caso o módulo não seja reconhecido ou a opção não esteja disponível, faça o *download* do drive 3g3AX-MX2-PRT-0C6A.gsd utilizando *Install Device Description Files* e instale o arquivo que será passado pelo professor. Selecione o módulo 3g3AX-MX2-PRT e *Add Device*.

Após adicionado o *device*, o módulo aparece no *My Network*.

Retorne à configuração clicando sobre o módulo 3G3AX-MX2-PRT e selecionando *Configuration* e ajuste *Station Address*: 2 (depende do

endereço do módulo configurado no inversor de frequência). Clicando em *Configuration* é possível definir diferentes tipos de operação associados a configurações do inversor de frequências. São exemplos: *Basic Data*, *Extended Data 1*, *Extended Data 2*, *PPO Type 1*, etc. Dependendo do modo selecionado as configurações a serem realizadas no inversor de frequência são diferentes. Utilizaremos o modo *Basic Data*. Clique e *Insert*. Em seguida, clique em DPV1 e depois desabilite o bit Enable DPV1 (obrigatório para permitir a comunicação). Clique em *Apply*.

Retornando para o CJ1W-PRM21, selecione *Slave Area* e verifique os endereços ajustados em *Address*. Os endereços da PROFIBUS podem ser ajustados para não produzir conflito com a Rede Device Net. Por exemplo, em *Output Allocation* o *Start Address* 4200 e 4400 podem ser ajustado no lugar de 3200 e 3400. Em *Input Allocation* o *Start Address* 4300 e 4500 pode ser ajustado no lugar de 3300 e 3500. Não se esquecer de clicar em *Save* após os ajustes. Depois salve o arquivo de configuração (neste momento o asterisco que aparecia na frente do módulo “desaparece”).

Com esse ajuste os bits programação do CLP estarão disponíveis nos endereços 4200, 4201 e 4202, sendo que o bit 1 do endereço 4200 é responsável por habilitar a comunicação entre o CLP e o módulo Profibus, enquanto a word em 4201 permite enviar o valor da frequência desejada indo de 0 a 60 Hz (enviando valores respectivamente de 0000 a 6000), como será mostrado na Figura 9.6.

Na barra de ferramentas selecione *Device / Go Online*. Selecione o PRM21 e *Download Parameters to Device* (ou utilize o ícone).

No inversor de frequência, o bit BF (*Bus Fail*) tem que estar apagado indicando que a comunicação com o módulo Profibus DP está OK. Caso contrário, se aceso o LED vermelho, indica que houve falha na comunicação.

Para verificar a configuração clique em *Device / Diagnosis* e selecione *Automatic*. Em *Diagnosis* é possível verificar se houve alguma falha na comunicação ou erro de configuração. Verifique o status em *Master Status* e depois em *Slave Status*. Se necessário selecione STOP e depois OPERATE. Pode ser necessário desligar o disjuntor também para reinicializar o inversor de frequência.

Nesse ponto a configuração do módulo Profibus e do inversor estão finalizadas.

Mais detalhes sobre os modos de configuração podem ser vistos no Manual do módulo disponível em:

https://assets.omron.eu/downloads/manual/en/v2/i111e_mx2_rx_profibus_option_board_users_manual_en.pdf

No Apêndice C desse manual, na página 79 são mostrados os modos de operação e o mapeamento do conjunto de words utilizadas na configuração (Word Offset). Para o caso do *Basic Data*, serão utilizados três words na configuração, sendo que a primeira define o *status* do inversor, a segunda define a frequência de operação do inversor (output frequency) e a terceira a referência de torque de saída. Note que para trabalhar nesse modo alguns bits configuração devem ser setados no inversor. Os três bits utilizados na configuração da primeira word permitem habilitar o inversor, ou inverter o sentido de rotação.

De modo similar, para os outros modos de operação, há um Word Offset disponível e bits a serem configurados no inversor.

Adicionalmente outros bits de configuração a serem ajustados estão apresentados na seção 3 do manual, a partir da página 23. Os bits que devem ser configurados no inversor de frequência para trabalhar no modo automático e *Basic Data* foram apresentados na página anterior deste documento.

Exemplo de Aplicação em Ladder

Para demonstrar o funcionamento da aplicação o professor irá utilizar um programa desenvolvido no software CX-One. O código do programa é apresentado na Figura 9.6. Observa-se no programa que, ao acionar o botão 0.00, o inversor de frequência, conectado no endereço 4200 da Rede PROFIBUS DP, será habilitado. Assim, caso o motor de indução esteja conectado na saída do inversor de frequência e o mesmo tenha sido conectado à rede trifásica, então será possível ajustar o valor de frequência utilizada como referência para o ajuste da velocidade. No exemplo, podem ser ajustados os valores 1000 ou 3000, os quais serão

automaticamente transferidos para o inversor de frequência respectivamente com o acionamento das chaves 0.01 ou 0.02. Note que nesses casos os valores ajustados serão mostrados no display do inversor de frequência e o motor de indução trifásico será acionado em velocidades proporcionais a esses valores. Observa-se também que, caso nenhuma das chaves for acionada, ou mesmo se ambas chaves forem acionadas, a velocidade será mantida, dependendo do último valor ajustado no inversor de frequência.

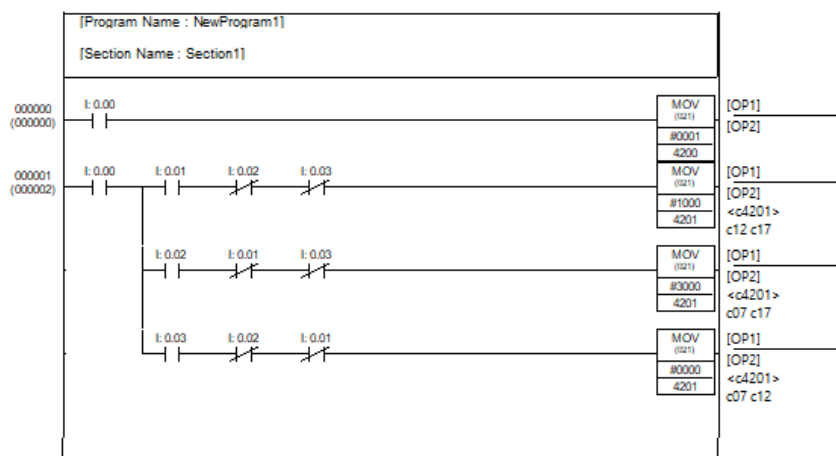


Figura 9.6 – Programa desenvolvido para CLP Omron em aplicação utilizando inversor de frequência e rede PROFIBUS DP

9.3 – Rede DeviceNet.

Nessa seção pretende-se demonstrar a configuração e montagem do CLP CJ2M para conexão de dispositivos utilizando redes DeviceNet.

A Figura 9.7 mostra um módulo DeviceNet e sua conexão em um CLP. Note que a comunicação é realizada por quatro fios, sendo dois fios para alimentação (cabos vermelho e preto) e dois fios para a transmissão

bidirecional. A Figura 9.8 ilustra a conexão de um módulo DeviceNet no CLP Omron.

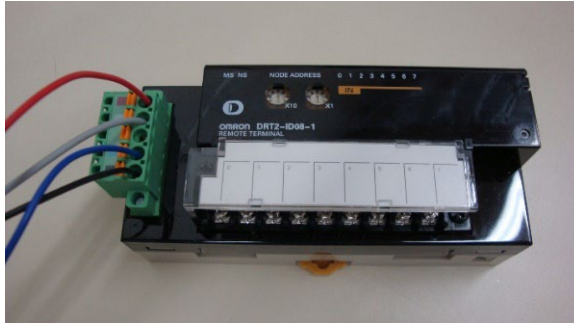


Figura 9.7 – Módulo DeviceNet DRT2-ID08

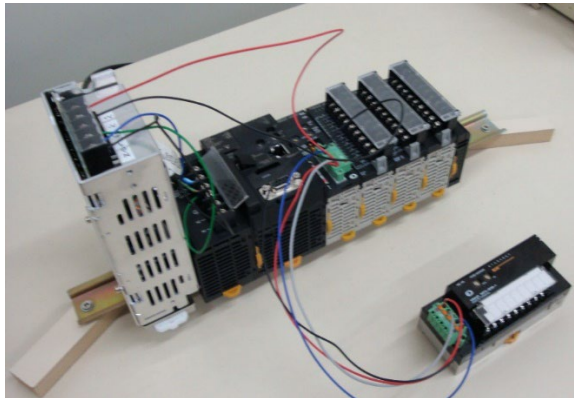


Figura 9.8 – Conexão do módulo DeviceNet DRT2-ID08 no CLP Omron

Configuração de Rede DeviceNet

O processo de instalação e configuração de uma rede DeviceNet é detalhado a seguir:

- verifique se os módulos estão conectados em 24 V (cabos vermelho e preto) e se os cabos de transmissão estão adequadamente instalados conforme Figura 9.7;

- identifique os endereços ajustados manualmente no módulo (exemplo: 0 e 1); os módulos devem assumir endereços diferentes;

- ligar o computador e abrir o software *CX-One Programmer*;

- abrir novo projeto e fazer as configurações iniciais selecionando tipo de equipamento (*device*) CJ2M, em *Settings CPU31* e *Network Type USB*;

- conectar, colocar no modo *Program* e em *IO Table and Unit Setup*, clicar em *Create Registered Table*;

- selecione o modulo DRM21 (modulo de rede Device Net);

- coloque o PLC no modo *Offline*;

- selecione *Start Special Application* e *Start Only*;

- automaticamente abre-se o software *CX-One Integrator*;

- selecione a opção *Network / Auto On-Line* (ícone) (dará erro se no CX-One o PLC ainda estiver no modo *On Line*);

- selecione USB e depois conecte;

- reconheça o cartão Device Net (clicar com botão do mouse direto sobre o cartão)

- selecione a opção *Connect* para criar a rede Device Net, selecione a opção *Network Structure and Parameter fo Each Component* e transfira a rede para o PC; irá aparecer a tela ilustrada na Figura 9.9, a qual mostra os dispositivos utilizados na rede e seus respectivos endereços; dependendo do número de dispositivos disponíveis poderá aparecer o módulo de entrada DRT2 – ID08-1, de saída DRT2 – OD08-1 bem como o módulo CJ1W-DRM21, que gerencia a rede; deve aparecer no PLC: sem erro, nó 63;

- para identificar os endereços dos módulos de entrada e saída e verificar se foram reconhecidos, faça dois cliques sobre o cartão DRM21, na rede mostrada e selecione *IO Allocation Out* ou *IO Allocation In*; é exibido o endereço do cartão (exemplo: 3200:Bit); modifique os endereços 3200 e 3300 respectivamente para 200 e 300 para evitar conflitos de endereço com a Rede Profibus; para isso clique em *Setup*; após realizar essa alteração atualizar no CLP usando o botão *Download*;

Cap. 9 – Redes Industriais

- feito isso a rede está preparada para utilização, para isso coloque o *CX-One Integrator* no modo *Offline* e retorne ao *CX-One Programmer*;
- para testar o módulo *DeviceNet*, com o PLC *On Line*, digite ALT 3; abre-se uma janela onde é possível testar dispositivos de entrada ou saída; para isso no campo Address digite o endereço do cartão (exemplo: 3200, 3201 ou 4200);
- clique duas vezes sobre a linha do endereço para abrir a tela de “Edit Dialog”; é possível entrar diretamente com dados de saída no formato hexadecimal (exemplo: #FFFF) ou abrir a tela para selecionar bit a bit (exemplo: *force set*).

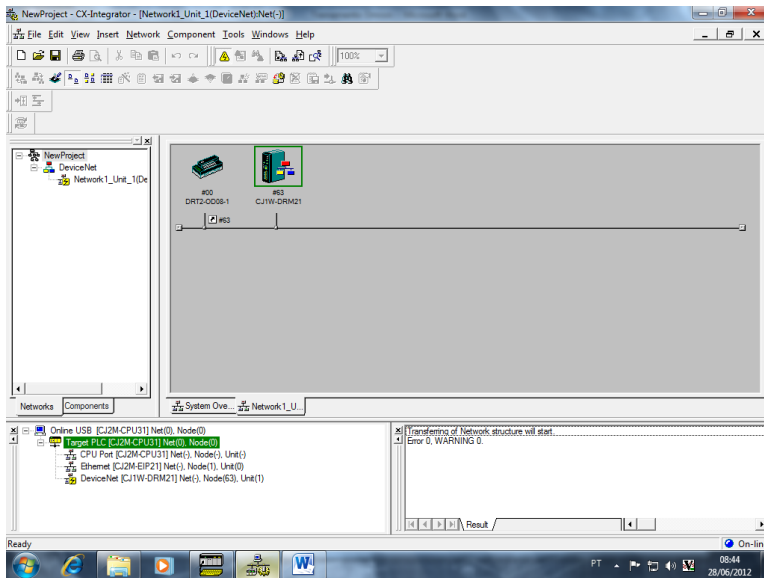


Figura 9.9 – Tela do software CX-Integrator após criação de uma rede DeviceNet

9.4 – Exemplo: Uma Aplicação Utilizando Rede DeviceNet.

Considere a rede DeviceNet criada no PLC Omron.

Desenvolva uma aplicação para acionar as saídas do módulo DeviceNet DRT2 – OD08 (Figura 9,10) utilizando a lógica apresentada a seguir:

- a aplicação deve funcionar se um *push-button* (*Start*) for acionado;
- após reconhecer o acionamento do botão *Start*, verifique as chaves 1 e 2;
- se a chave 1 for acionada, reproduza nos quatro bits menos significativos da saída do módulo DeviceNet a combinação produzida por oito chaves; os quatro bits mais significativos devem ser mantidos em zero;
- se a chave 2 for acionada (e mantida dessa forma), reproduza na saída do módulo DeviceNet um contador de 8 bits, com pulsos sendo incrementados por um *push-button* de incremento; lembre-se que no momento que a chave 2 for acionada, o contador deve começar a funcionar, iniciando-se com contagem nula.

Cap. 9 – Redes Industriais

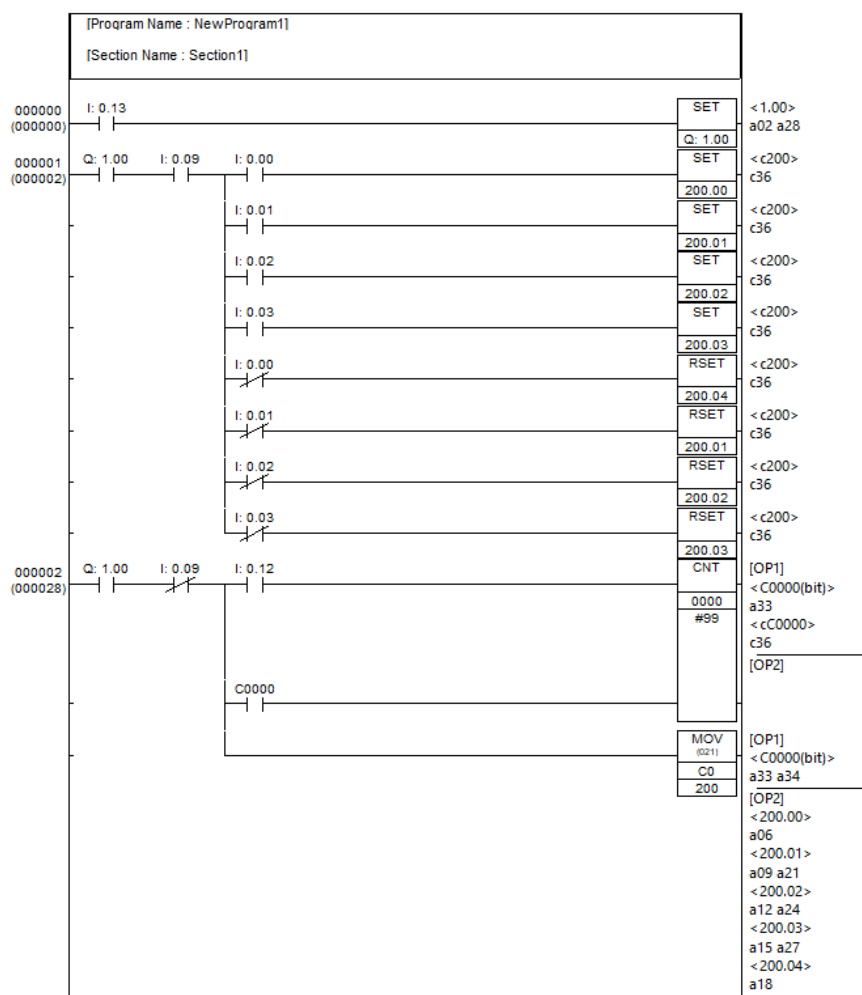


Figura 9.10 – Aplicação Utilizando Rede DeviceNet

10 – Sistemas de Supervisão e Aquisição de Dados

10.1 – Sistema SCADA.

Sistemas de Supervisão e Aquisição de Dados, ou abreviadamente SCADA (*Supervisory Control and Data Acquisition*) são sistemas que utilizam software para monitorar e supervisionar as variáveis e os dispositivos de sistema de controle conectados através de controladores específicos.

Alguns exemplos de softwares SCADA são: Indusoft Web Studio (IWS), Elipse SCADA, Elipse E3, WinCC, Wonderware InTouch, iFix, entre outros.

Por meio dos sistemas de supervisão o usuário pode monitorar e controlar partes ou todo um processo industrial diretamente por um ou mais terminais. Adicionalmente os sistemas de supervisão permitem comunicação utilizando diversos protocolos industriais bem como a integração com sistemas gerenciais, bancos de dados e softwares de planejamento como MES, ERP, SAP e etc.

Os benefícios incluem, além da flexibilidade e versatilidade, melhoria da qualidade dos processos produtivos por meio da determinação dos níveis ótimos de trabalho, redução de custos operacionais, maior agilidade e eficiência de operação, diagnóstico e correção de eventuais falhas.

As principais funcionalidades são importantes em softwares SCADA:

- a) possibilidade de criação de tela de **sinóticos**;
- b) geração de **alarmes** de processo *on line* e **relatórios** com **histórico** de variáveis de processo;
- c) geração de **gráficos históricos** e **banco de dados**;
- d) comunicação com controladores de campo (CLPs) por meio de um protocolo de forma a monitorar variáveis de processo.

A tela de **sinóticos** é a tela principal do supervisório que permite monitorar um processo industrial por meio de objetos que simulam as características do sistema. A tela de **alarmes** permite gerar avisos ao usuário do sistema, na forma de planilhas e/ou animações na tela, quando uma variável ou condição de processo de produção está fora dos valores previstos. A tela de **relatórios** lista um **histórico** com os **alarmes** ocorridos durante uma faixa de tempo. Permite também verificar os usuários que acessaram o sistema ou alteraram algum parâmetro (**relatório de acesso**) ou a alteração de variáveis ao decorrer do tempo (**relatório de variáveis**). **Gráficos históricos e de tendências** permitem analisar graficamente o comportamento de variáveis do sistema.

Para a comunicação com CLPs tipicamente utiliza-se como meio físico os padrões RS232 (até 12 metros), RS485 (até 1200 metros, sem repetidores) ou Ethernet (até 100 metros entre segmentos). Para distâncias elevadas utiliza-se fibra óptica.

Os protocolos utilizados na comunicação podem ser o Foundation Fieldbus, Profibus, Profibus-DP, Profibus-PA, Profibus-MS, ModBus, DeviceNet, entre outros.

Neste capítulo o objetivo é desenvolver as funções a), b) e c) utilizando o IWS (*Indusoft Web Studio*).

10.2 – Indusoft Web Studio.

O software IWS (*Indusoft Web Studio*) é um software SCADA desenvolvido pela empresa de origem alemã Indusoft. O IWS é uma ferramenta integrada que explora as características dos sistemas operacionais Windows e do Windows CE de forma a permitir a construção de todas as características de aplicações SCADA e HMI (*Human-Machine Interface*).

Algumas funcionalidades do IWS são:

- interface gráfica completa para o desenvolvimento de sistemas de automação incluindo biblioteca com mais de 100 símbolos e objetos dinâmicos tais como medidores, *sliders*, veículos, motores, válvulas, entre outros;

- possui mais de 150 drivers de comunicação com diferentes fabricantes de dispositivos (Allen-Bradley, Siemens, GE-Fanuc, etc.);
- expressões lógicas e linguagem de *scripting* com mais de 200 funções;
- editor de software interessante que permite visualizar a dinâmica do processo industrial por meio de *Web browser*;
- permite troca de dados com dispositivos *wireless* e móveis;
- fácil integração com outros aplicativos Windows, como o Word e o Excel;
- interface com outros sistemas como o Java e o Visual Basic;
- sistema de alarmes que permite o envio de mensagens para vários destinos, como as telas, e-mail, *browser* e arquivos;
- permite utilização de objetos com os principais padrões de software industrial tais como o OPC, DDE, ODBC, XML e ActiveX.

O ambiente de desenvolvimento (IDE) do Indusoft Web Studio é apresentado na Figura 10.1. A barra de status do IDE é apresentada na Figura 10.2. O Explorador de Projeto do IDE está apresentado na Figura 10.3. Por meio do explorador é possível o acesso às abas Global, Gráficos, Tarefas e Comunicação. Na aba Global é possível inserir novos *tags*, classes, procedimentos, bem como consultar *tags* de Sistema e Habilitar/Desabilitar *Logs* de Eventos e Planilhas de Tradução da Aplicação. Na aba Gráficos é possível configurar telas, grupos de telas e páginas Web, além de acessar biblioteca de Símbolos para a aplicação. Na aba Tarefas é possível configurar as planilhas executadas no aplicativo. Por fim, na aba Comunicação tem-se acesso à configuração de mais de 250 drivers de comunicação, podendo utilizar a comunicação OPC, TCP/IP (entre Indusoft) e DDE.

Cap. 10 – Sistemas de Supervisão e Aquisição de Dados

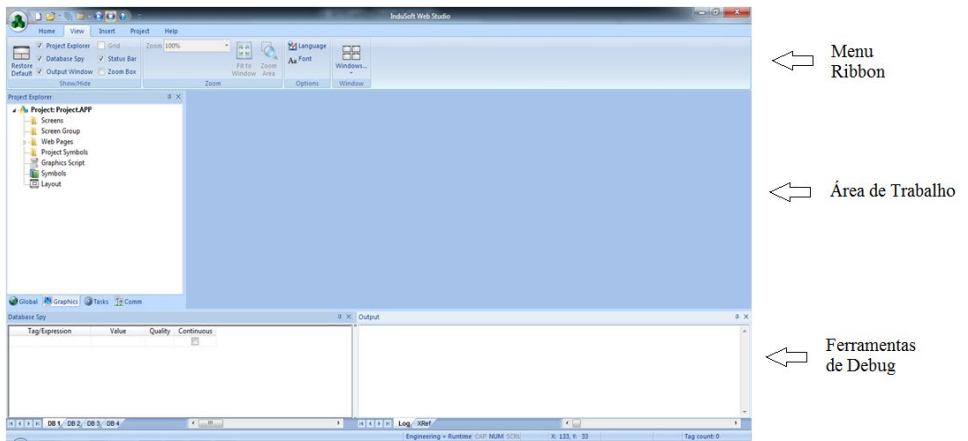


Figura 10.1 – Ambiente de Desenvolvimento (IDE) do IWS

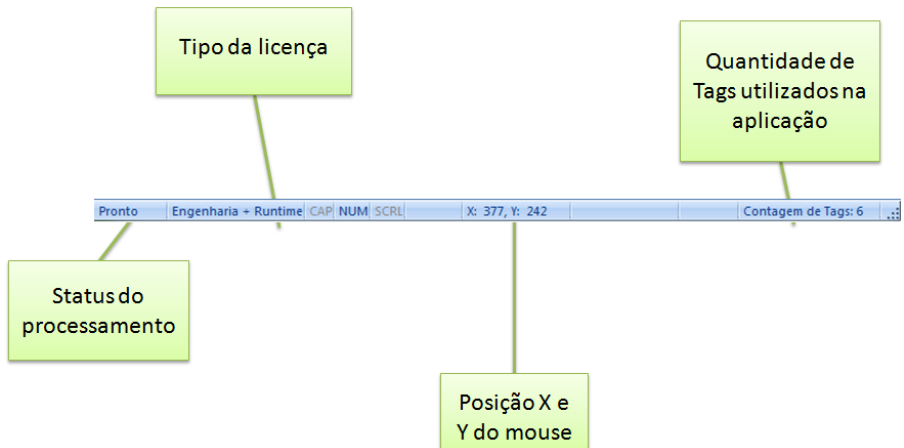


Figura 10.2 – Barra de Status do Ambiente de Desenvolvimento do IWS

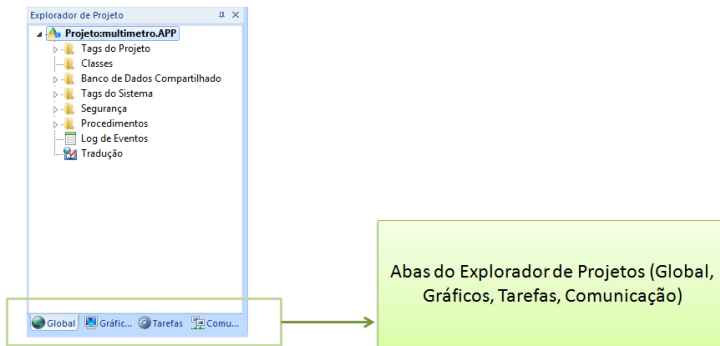


Figura 10.3 – Explorador de Projeto do Ambiente de Desenvolvimento do IWS

10.3 – Exemplo: Aplicação de Controle em Tanques de Mistura.

No experimento utilizaremos o software IWS (Indusoft Web Studio), versão 7.1 para desenvolver o projeto de um supervisório para controle e segurança de tanques de mistura como ilustrado na Figura 10.4. As especificações desejadas para o projeto são:

- o aplicativo deverá controlar a produção, desde a seleção da receita do produto a ser produzido, até a estocagem do produto final;
- o supervisório deverá armazenar os dados de nível, temperatura, vazão e pressão dos tanques (2 tanques de mistura e 1 de estocagem);
- deverá gerar relatórios e gráficos de processo OnLine e Histórico;
- deverá informar alarmes de processo (ALTO e BAIXO) OnLine e Histórico (para Temperatura e Nível de cada tanque);
- as variáveis de campo poderão ser simuladas, tornando-se desnecessária a comunicação com um CLP.

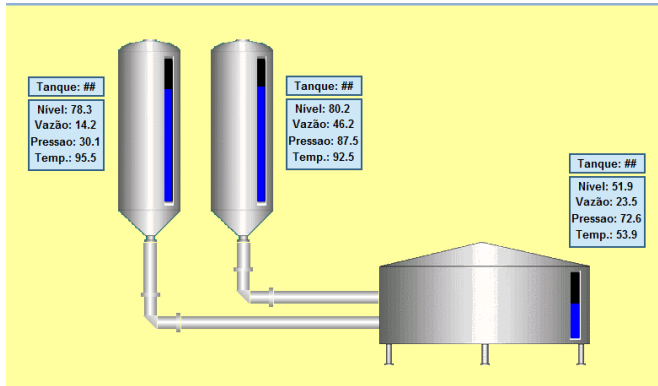


Figura 10.4 – Tela de Sinótico de um Sistema de Controle de Tanques de Mistura

Para isso, devem ser obedecidas as instruções apresentadas sequencialmente abaixo:

a) criação de novo projeto.

Na aba superior ou no símbolo do Indusoft, criar novo projeto.

- Novo projeto: escolher nome sem espaço ou acentos; salvar a tela na pasta disponível no computador (exemplo: T:/ ou C:/temp);
- em plataforma alvo selecione a opção Windows e Windows Local Interface (essa é a opção de licença Standard do IWS (Local Interface) que apresenta 1500 Tags);
- resolução: 1440 x 900 (ou menos, caso contrário o projeto completo não poderá ser mostrado na tela do computador) – exemplo: 1366x768 ou 1280x800;
- será criado um projeto em branco.

Após a criação do projeto, o nome aparecerá no Explorador de Projeto.

b) criação de telas.

Para criarmos novas telas em um projeto, iremos na aba “Gráficos” do “Explorador de projetos”, como mostrado na Figura 10.5 e então clicamos com o botão direito do mouse sobre o diretório “Telas” da “Árvore do projeto”. Então clicamos com o botão esquerdo do mouse sobre a opção “Inserir”. As características das telas podem ser escolhidas conforme ilustrado na Figura 10.6 onde deve-se digitar em “Descrição:” o nome da tela. Crie 3 telas iniciais chamadas SUPERIOR, INFERIOR e MODELO. Selecione cada tela, uma de cada vez, e clicando sobre a tela com o mouse selecione a cor. Modifique as cores de fundo de cada tela deixando a tela SUPERIOR e INFERIOR com tons escuros, e a tela MODELO com um tom mais claro. Para isso, abra primeiramente a tela “SUPERIOR” e clicando sobre “Disposição” no Explorador de Projetos, posicione e redimensione a tela. Clicando sobre a tela com o botão direito do mouse escolha a cor de fundo. Realize a mesma operação nas telas SUPERIOR E INFERIOR. As três telas resultantes podem ficar dispostas como mostrado na Figura 10.7. Após concluir essas configurações, clique sobre o identificador de cada tela, acionando o botão direito do mouse, salve a tela em arquivo (na pasta onde o projeto foi salvo e dentro do diretório *Screen*).

No Explorador de Projeto, crie um “grupo de telas” com as telas SUPERIOR, INFERIOR e MODELO e chame-o de INICIAL. Com o botão direito do mouse configure para que o INICIAL seja a “tela inicial” de nossa aplicação.

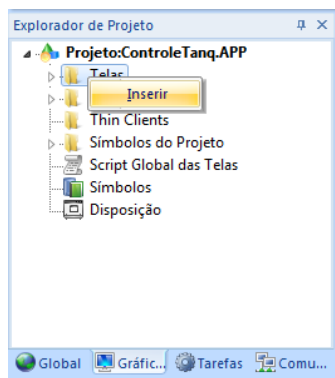


Figura 10.5 – Criação de Telas

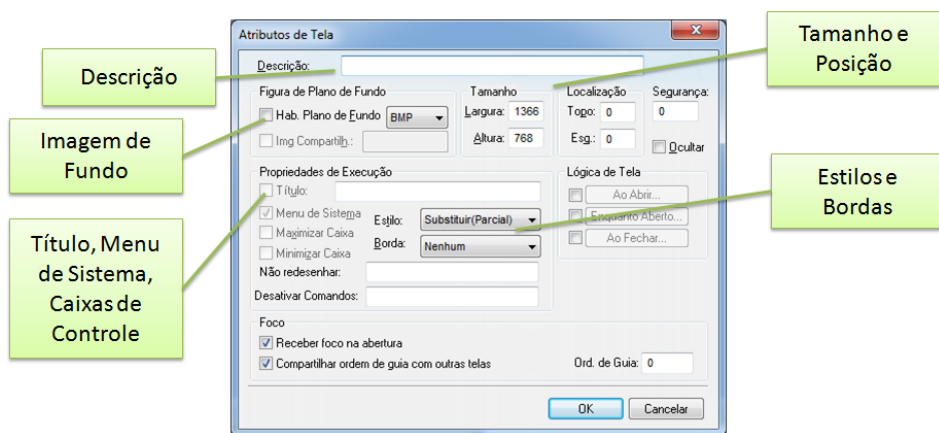


Figura 10.6 – Características das Telas

Cap. 10 – Sistemas de Supervisão e Aquisição de Dados

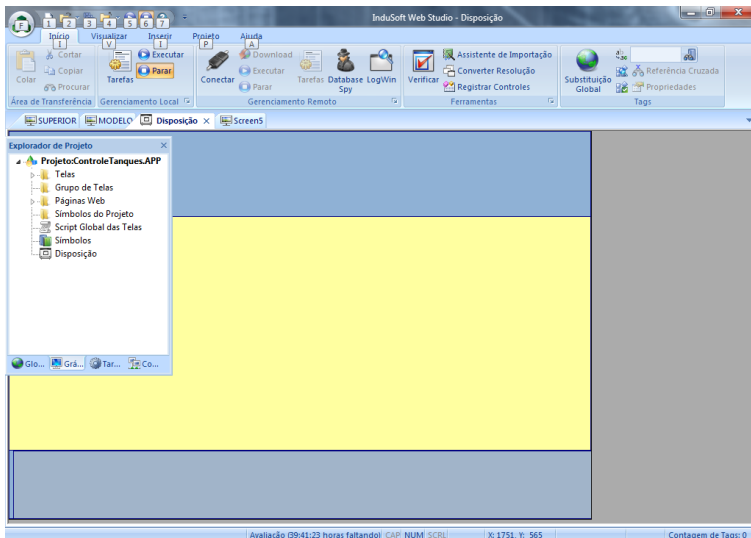


Figura 10.7 – Disposição das Telas Superior, Modelo e Inferior

c) inserindo botões nas telas.

Inicialmente selecione a tela INFERIOR. Para inserir um botão nessa tela basta clicar sobre a aba “Gráficos” do menu “Ribbon” na área denominada “Objetos Ativos e arraste o mouse sobre a tela (Figura 10.8).



Figura 10.8 – Objetos Ativos

Clicando duas vezes sobre o objeto inserido na tela, teremos acesso às “propriedades” daquele objeto específico, podendo modificar o tipo de objeto, alterar o texto inserido, alterar cor, etc.

Nessa etapa iremos criar os seguintes botões:

- na tela “INFERIOR”, crie 4 Botões e mude a opção “Legenda” para SINOTICO1, SINÓTICO2, ALARME e TREND. Pretende-se dessa forma criar botões para selecionar as telas que iremos mostrar no supervisório, respectivamente como mostrado na Figura 10.9; para mover os botões clique sobre a tela e desabilite “Desativar Arrasto”; nessa seleção podem ser consideradas telas como SINÓTICO, ALARME, TREND (gráficos de tendência), EVENTO, RECEITA, RELATÓRIO, BANCO DE DADOS e SAIR (visando parar a execução do supervisório);

- abra a tela “MODELO” e então clique em arquivo (Logo Indusoft), clique em “Salvar Como” e salve essa tela 4 vezes como SINOTICO1, SINOTICO2, ALARME e TREND, respectivamente;

- clique sobre a tela “INFERIOR” e em cada botão selecione com o mouse a opção “Propriedades” e altere a propriedade “COMANDO” de cada um dos botões para o tipo “Abrir Tela” e configure para abrir a tela com mesmo nome da “Legenda” do Botão.

Clique em “Comando” e depois em “Tipo” e escolha a opção “Abrir tela”. Selecione em “Abrir tela” o nome da tela associada. Assim, cada vez que o botão for acionado numa aplicação ele deverá abrir a tela correspondente.

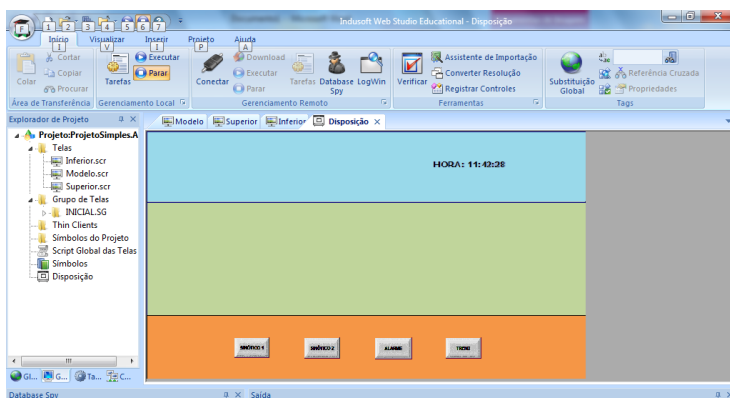


Figura 10.9 – Grupo de Telas Inicial com Telas SUPERIOR, MODELO E INFERIOR, com Botões na Tela INFERIOR.

d) inserindo texto com hora atual.

Selecione a tela SUPERIOR. Clique sobre a aba “Gráficos” do menu “Ribbon” na área denominada “Objetos Ativos” (Figura 10.7) e selecione “Texto” arrastando-o para a tela. Digite o texto “Hora:” Clique sobre o objeto, selecione “Texto com Link de Dados” e em Tag/Expressão localize a tag interna “Time”.

Inserir propriedades do objeto: dimensionar automaticamente ou voltar para texto e inserir o tamanho desejado com 6 dígitos (basta inserir #####).

e) criação de objetos de dados.

O software IWS possui três Objetos de Dados (Figura 10.10). São eles: Alarme/Evento, Tendência e Grid. Inicialmente iremos criar um objeto na tela de alarme. Para isso, na aba “Gráficos” selecione a opção Alarme/Evento e insira na tela ALARME, movendo o mouse.



Figura 10.10 – Objetos de Dados

Seguindo o procedimento insira os seguintes objetos de dados:

- na tela “ALARME”, crie 2 Objetos de Dados de Alarme/Evento;
- na tela “TREND”, crie um Objeto de Dado de Tendência;

Neste ponto, um teste da aplicação desenvolvida até o momento pode ser feito. Para isso, clique sobre o ícone “Executar” e as telas serão dispostas como foram preparadas, sendo possível excursionar em cada tela conforme o botão acionado. Para retornar, clique em ALT TAB e pare a execução.

f) criação de tags.

Nessa etapa serão criados tags para identificar variáveis como nível, pressão, vazão e temperatura.

Existem vários tipos de tags (Figura 10.11): booleana (b), inteira (n), real (r), string até 1024 caracteres (s) e classe (c). Classe são tags que contém vários membros de vários tipos. Exemplo: uma classe “Tanque” pode conter as tags “nível”, “pressão” “vazão” e “temperatura”.

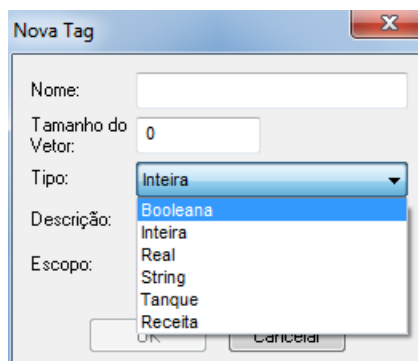


Figura 10.11 – Tags

Crie a classe “Tanque” clicando na aba “Global” / “Classes” e com o botão direito do mouse “Inserir Classe” (Figura 10.12). Em seguida crie quatro membros para essa classe, sendo eles:

- rNível – Tipo: Real – Descrição: Nível dos tanques;
- rVazao – Tipo: Real – Descrição: Vazão dos tanques;
- rPressao – Tipo: Real – Descrição: Pressão dos tanques;

- rTemp – Tipo: Real – Descrição: Temperatura dos tanques.

g) criação de vetores.

A utilização de vetores (Figura 10.13) permite uma fácil organização de dados, além de permitir a utilização de *scripts*.

Crie uma tag do tipo classe Tanque (que foi criado anteriormente) com as seguintes especificações:

- nome da Tag: cTanque;
- tamanho do vetor: 3;
- tipo: Tanque;
- descrição: “Tanque da aplicação”;
- escopo: servidor .

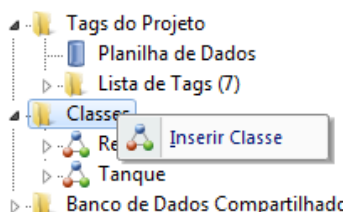


Figura 10.12 – Inserindo Classe

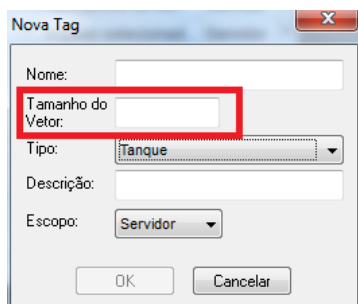


Figura 10.13 – Inserindo vetor para Tags

h) criação de tela de sinótico.

Nessa etapa será desenvolvida a tela de sinótico a qual apresenta uma visão geral da aplicação. Criar sinótico com dois tanques de mistura e um de estocagem (veja a Figura 10.14). Para isso serão utilizados quatro símbolos de tanques, símbolos de canos ligando os tanques e painéis com dados dos tanques. Os tanques podem ser localizados na pasta “Tank”. Os canos podem ser localizados na pasta “Pipes”.

Para inserir símbolos selecione a aba “Gráficos”, selecione “Symbols” clique duas vezes sobre o símbolo que quiser e coloque na tela.

Para tirar a grade e facilitar a montagem do sinótico, clique na tela, acionando o botão direito do mouse e desabilite a opção “Configuração de Grade” e “Alinhar à Grade”.

Para alterar o tamanho, clique no símbolo e desabilite a opção “Usar tamanho vinculado”.

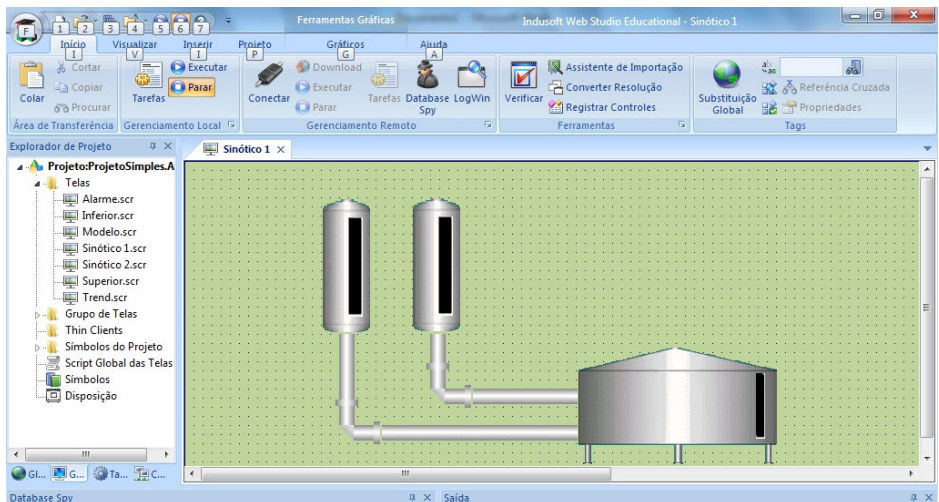


Figura 10.14 – Criação de tela SINÓTICO

Clique no símbolo para ajustar as suas propriedades. Para realizar essa ação o símbolo deve ser configurado como Linked Symbol. Assim,

clicando em propriedade podemos visualizar a tela apresentada na Figura 10.15. Coloque na propriedade “TagLevel” de cada um dos três tanques de mistura os seguintes valores: cTanque[0].rNivel e cTanque[1].rNivel. No tanque de mistura, coloque na propriedade “TagLevel” o valor: cTanque[2].rNivel.

Além dos símbolos já inclusos, é possível criar um símbolo personalizado. Criaremos um painel de monitoramento de variáveis dos tanques como mostrado nas Figuras 10.4 e 10.16. Para isso:

- crie dois retângulos, um como cabeçalho e outro, maior, como corpo; retângulos podem ser obtidos na aba Gráficos > Formas;
- utilizando a aba Gráficos > Texto, coloque um texto como “Tanque: ##” no cabeçalho do painel que queremos criar;
- no corpo do painel, coloque quatro objetos de textos, que são: “Nível: #####”, “Pressão: #####”, “Temp.: #####”, “Vazão: #####”.

Para configurar as propriedades associadas aos símbolos:

- habilite a opção que permite entrada e saída de dados em cada texto;
- selecione “Texto com Link de Dados”;
- para cada objeto, seguiremos o padrão para identificar o Tag/Expressão: “#Categoria.propriedade:valorPadrão”;
- para o texto “Tanque: ##” coloque “#Tanque.NdoTanque:0”;
- para o texto “Nível: #####”, coloque “#Dados.Nivel:”;
- para o texto “Pressão: #####”, coloque “#Dados.Pressao:”;
- para o texto “Temp.: #####”, coloque “#Dados.Temp:”;
- para o texto “Vazão: #####”, coloque “#Dados.Vazao:”.

Não se esqueça de preencher as duas categorias.

Para converter painel ou qualquer objeto em símbolos:

- selecione o objeto do símbolo (nesse caso iremos criar um símbolo com o painel completo);
- clique com o direito do mouse;

- selecione “Criar Linked Symbol”; o símbolo criado será adicionado à biblioteca do IWS com o nome que escolhermos (por exemplo: painel.sym).

Coloque um símbolo de painel para cada tanque da tela SINOTICO.SCR (total de três painéis) e associe as propriedades com suas tags de classe Tanque conforme Figuras 10.16 e 10.17.

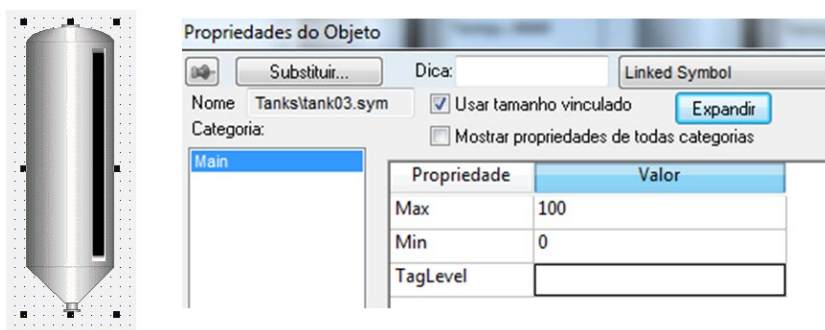


Figura 10.15 – Inserindo vetor para Tags na aplicação

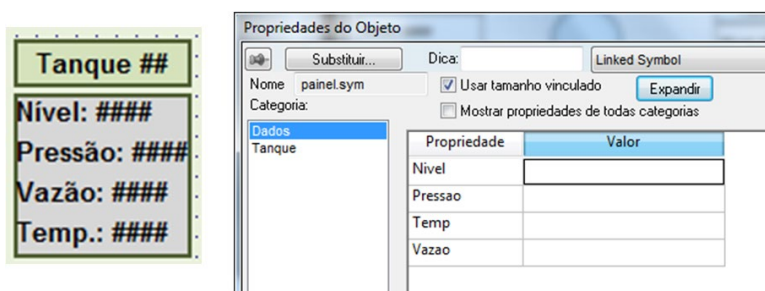


Figura 10.16 – Criando Símbolos para as Variáveis

Cap. 10 – Sistemas de Supervisão e Aquisição de Dados

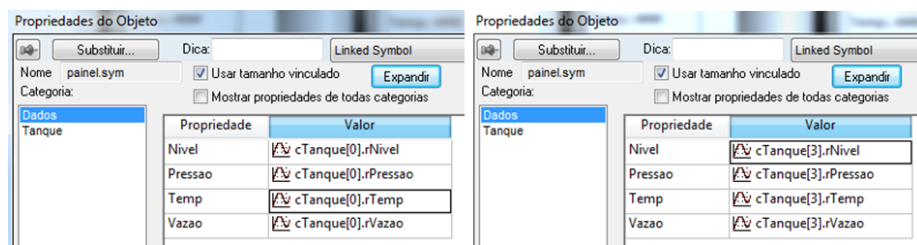


Figura 10.17 – Configurando as Propriedades do Objeto

Para simular valores das variáveis serão usadas as funções Agenda, Rand e Checkbox.

A função “Inserir Agenda” pode ser obtida em Explorador de Projetos > Tarefas > Agenda (Figura 10.18). A opção calendário permite atualizar valores a cada data selecionada. A opção relógio permite agendamento de hora para atualização. Podemos selecionar o campo Alterar para Segundo, por exemplo.

A função Rand gera valores aleatórios entre 0 e 1 e pode ser obtida em Explorador de Projetos / Tarefas / Agenda / Expressão (Figura 10.19). A função Check Box permite utilizar um teste para habilitar ou não uma simulação. Isso por ser feito, por exemplo, na tela “SUPERIOR” criando uma checkbox com as propriedades de objeto para habilitar ou não uma simulação, por meio de variável booleana (Figura 10.20). É necessário setar condição de desligamento do evento de agenda em Explorador de projetos > Tarefas > Agenda > Desativar (Figura 10.21).

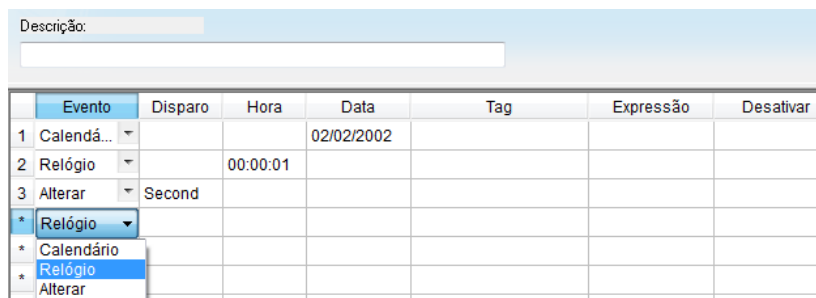


Figura 10.18 – Agenda

	Evento	Disparo	Hora	Data	Tag	Expressão
1	Relógio		00:00:01		cTanque[0].rNivel	\$Rand()*100

Figura 10.19 – Função Rand



Figura 10.20 – Função Check Box

	Evento	Disparo	Hora	Data	Tag	Expressão	Desativar
1	Relógio		00:00:01		cTanque[0].rNivel	\$Rand()*100	bSimulacao = 0

Figura 10.21 – Desligamento de Agenda

Utilizando essas funções podemos criar os seguintes eventos de AGENDA para os quatro membros do vetor cTanque e seus quatro membros, setando, como mostrado na Figura 10.22:

- Evento: Relógio;
- Hora: 00:00:01;
- Tag: cTanque[3].rTemp e/ou cTanque[1].rNivel...
- Expressão: \$Rand()*100;
- Desativar: bSimulacao = 0;

Cap. 10 – Sistemas de Supervisão e Aquisição de Dados

Descrição:							
Simulação							
	Evento	Disparo	Hora	Data	Tag	Expressão	Desativar
1	Relógio		00:00:01		cTanque[0].rNivel	\$Rand()*100	bSimulacao=0
2	Relógio		00:00:01		cTanque[1].rNivel	\$Rand()*100	bSimulacao=0
3	Relógio		00:00:01		cTanque[2].rNivel	\$Rand()*100	bSimulacao=0
4	Relógio		00:00:01		cTanque[0].rPressao	\$Rand()*100	bSimulacao=0
5	Relógio		00:00:01		cTanque[1].rPressao	\$Rand()*100	bSimulacao=0
6	Relógio		00:00:01		cTanque[2].rPressao	\$Rand()*100	bSimulacao=0
7	Relógio		00:00:01		cTanque[0].rTemp	\$Rand()*100	bSimulacao=0
8	Relógio		00:00:01		cTanque[1].rTemp	\$Rand()*100	bSimulacao=0
9	Relógio		00:00:01		cTanque[2].rTemp	\$Rand()*100	bSimulacao=0
10	Relógio		00:00:01		cTanque[0].rVazao	\$Rand()*100	bSimulacao=0
11	Relógio		00:00:01		cTanque[1].rVazao	\$Rand()*100	bSimulacao=0
12	Relógio		00:00:01		cTanque[2].rVazao	\$Rand()*100	bSimulacao=0
13							
14							

Figura 10.22 – Configurações Finais de Agenda

No preenchimento será solicitada e criada a tag booleana bSimulacao.

Para trabalhar com a tag bSimulacao, vá para a aba SUPERIOR e insira uma caixa de seleção por meio da aba Gráficos > Caixa de Seleção; arraste para a tela, clique e escreva bsimulacao na tag. Pode mudar o nome de “Seleção” para “Habilita Simulação”. Salvar a Agenda produzida.

Nesse ponto é possível temos uma tela de sinótico pronta, como mostrado na Figura 10.23, inclusive com simulação. A continuidade do projeto permite obter tela de alarmes, tendências (TREND), eventos, receitas, relatórios, banco de dados, etc. Também é possível conectar-se ao CLP para obter informações do sistema em tempo real. Contudo esse não é o objetivo do exemplo proposto.

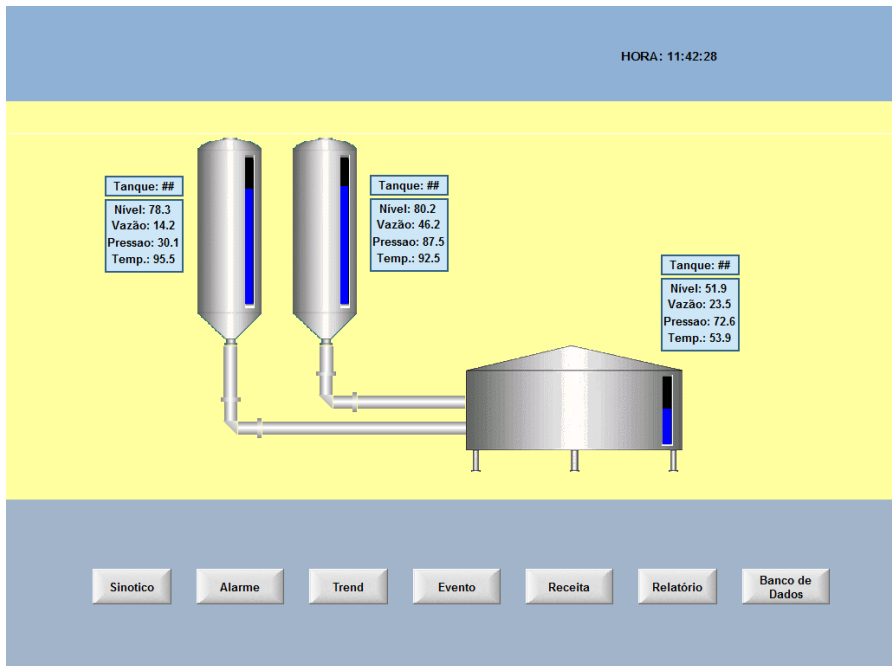


Figura 10.23 – Exemplo de Tela de Sinótico em Software Indusoft Web Studio

REFERÊNCIAS BIBLIOGRÁFICAS

ATTÍE, S. S. **Automação Hidráulica e Pneumática Empregando a Teoria de Sistemas**

a Eventos Discretos Dissertação de Mestrado, UFSC, 1998.

BATTEN JR, G. L. **Programmable Controlers – Hardware, Software and Applications**

Editora McGraw Hill, 1994.

BONACORSO, N. G.; NOLL, V. **Automação Eletropneumática** Editora Érica, São

Paulo, 1999.

CAPUANO, Francisco Gabriel; **Exercícios de Eletrônica Digital**. 3. ed. São Paulo, SP: Érica, 1991. 183 p.

ERCEGOVAC, Milos D; LANG, Tomás; MORENO, Jaime H. **Introdução aos Sistemas Digitais**. Porto Alegre, RS: Bookman, 2000. 453 p.

FARRER, H.; et al. **Programação Estruturada de Computadores: Pascal Estruturado**. Rio de Janeiro: LTC Editora, 3a. edição, 1999.

FIALHO, A. B. **Automação Pneumática – Projetos, Dimensionamento e Análise de**

Circuitos Editora Érica, São Paulo, 2003.

FONSECA, M. O., SEIXAS FILHO, C., e BOTTURA FILHO, J. A. **Aplicando a Norma IEC 61131 na Automação de Processos** ISA América do Sul, Distrito 4, São Paulo, 2008.

FREGNI, E. e SARAIVA, A. M. **Engenharia de Projeto Lógico Digital**. Editora Edgard Blucher, 1995.

- IDOETA, Ivan Valeije; CAPUANO, Francisco Gabriel. **Elementos de Eletrônica Digital**. 26. ed. São Paulo, SP: Érica, 1997.
- LUGLI, A. B.; SANTOS, M. M. D. **Sistemas Fieldbus para Automação Industrial - DeviceNet, CANOpen, SDS e Ethernet**, 1a Ed., São Paulo, SP: Érica, 2009.
- LUGLI, A. B.; SANTOS, M. M. D. **Redes Industriais para Automação Industrial - AS-I, PROFIBUS e PROFINET**, 1a Ed., São Paulo, SP: Érica, 2010.
- MANO, M. Morris; KIME, Charles R. ***Logic and computer design fundamentals***. New Jersey : Prentice Hall, 2000, 652p.
- MIYAGI, P. E. **Controle Programável – Fundamentos do Controle de Sistemas a Eventos Discretos** Editora Edgard Blücher Ltda, São Paulo, 1996.
- MORAES, C. C. e CASTRUCCI, P. L. **Engenharia de Automação Industrial** Editora LTC, Rio de Janeiro, 2007.
- NATALE, F. **Automação Industrial** Editora Érica, São Paulo, 1997.
- SILVEIRA, P. R. e SANTOS, W. E. **Automação e Controle Discreto** Editora Érica Ltda, São Paulo, 1998.
- SOUZA, M. A. F.; SOARES, M. V.; GOMES, M. M.; CONCILIO, R. **Algoritmos e Lógica de Programação**. 1a. e 2a. edição, São Paulo: Cengage Learning, 2011.
- TAUB, Herbert; SCHILLING, Donald. **Eletrônica digital**. São Paulo, SP: McGraw-Hill, 1982. 582 p.

Referências Bibliográficas

TOCCI, Ronald J.; WIDMER, Neal S. **Sistemas digitais**: princípios e aplicações. São Paulo: Pearson/Prentice Hall, 2003, 753p.

UYEMURA, John P. **Sistemas digitais**: uma abordagem integrada. São Paulo, SP: Pioneira Thomson Learning, 2002, 433p.

Anexo.

Simbologia de Dispositivos Pneumáticos e Eletropneumáticos (Copyright by Festo Didactic)

Anexo

Simbologia Pneumática e Hidráulica				Simbologia Elétrica			
Atuadores		Válvulas Direcionais		Indicadores			
	Simple Ação retorno por força externa		2/2 vias normal fechada		Óptico		
	Simple Ação retorno por mola		3/2 vias normal fechada		Sonoro		
	Dupla Ação		3/2 vias normal aberta	Atuadores Eletromecânicos			
	Dupla Ação com amortecimento		4/2 vias				
	Dupla Ação com haste passante		5/2 vias				
	Dupla Ação Diferencial		4/3 vias centro A e B despressurizados				
	Dupla Ação sem haste		4/3 vias centro fechado		Relé com atraso na ativação		
Motores		Válvulas de Bloqueio		Contatos			
	com 1 direção de rotação		Válvula de Bloqueio		Nórmal Aberto		
	com 2 direções de rotação		Válvula de Simultaneidade (E)		Normal Fechado		
	de giro limitado		Válvula Alternadora (OU)		Comutador		
Conversão de Energia			Válvula Reguladora de Fluxo Unidirecional		N.A. acionado por botão pulsador		
	Compressor		Válvula de Escape Rápido		N.F. acionado por botão com trava		
	Bomba Hidrostática		Válvula de Registro		N.A. com atraso na ativação		
Unidade de Conservação		Válvulas de Pressão			N.A. com atraso na desativação		
	Filtro com dreno, Regulador de pressão e lubrificador		Válvula Limitadora de Pressão		N.A. acionado por rolete		
	Simbologia Simplificada		Válvula Reguladora de Pressão		Pressostato		
Tipos de Acionamento		Simbologia Lógica		Sensores			
Manual			Função And (E)		Sensor N.A. P.N.P.		
	Geral		Função OR (OU)		Sensor N.A. N.P.N.		
	Botão		Função Not (Negação)	Diversos			
	Alavanca		Função Bi-estável (Flip Flop)				
	Pedal	FESTO DIDACTIC Festo Automação Ltda. Rua Giuseppe Crespi, 76 (km 12,5 da Via Anchieta) Tel.: (011) 969-9500 Fax.: (011) 947-7290					
Elétrico							
	Solenóide		Fonte de Pressão		Reservatório		
Pressão			Silenciador		Secador		
	Piloto		Servo-piloto				

ISBN: 978-65-83488-02-2

CSL



9 786583 488022

INSTITUTO MAUÁ DE TECNOLOGIA



MAUÁ