



UNIVERSIDADE ESTADUAL PAULISTA

Pós-Graduação em Engenharia Elétrica

**“AUTOMAÇÃO INDUSTRIAL: UMA PROPOSTA
PARA ENSINO NO CURSO DE ENGENHARIA
ELÉTRICA”**

Jean Marcos de Souza Ribeiro
Engenheiro Eletricista - FEIS/UNESP

Ilha Solteira - SP

Universidade Estadual Paulista "Júlio de Mesquita Filho" - UNESP
Faculdade de Engenharia de Ilha Solteira - FEIS
Departamento de Engenharia Elétrica - DEE

Curso de Pós-Graduação em Engenharia Elétrica
Stricto Sensu
Área de Automação

"AUTOMAÇÃO INDUSTRIAL: UMA PROPOSTA PARA ENSINO NO CURSO DE ENGENHARIA ELÉTRICA"

Proc. 071/2002 - NPD 152/02

UNESP - "CAMPUS DE ILHA SOLTEIRA"	
SERVIÇO TÉCN. DE BIBLIOTECA E DOCUMENTAÇÃO	
DATA DE CHEGADA	DATA DE TOMBO
02.12.02	31.12.02
RECEBIDO POR	TOMBO
<i>ailza</i>	<i>Tr. 1329</i>
AQUISIÇÃO	CLASSIFICAÇÃO
<i>Doação autor R\$ 10,00</i>	<i>R484a</i>

1210001329



"Dissertação submetida à Faculdade de Engenharia de Ilha Solteira - UNESP, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica, junto ao Curso de Pós-Graduação em Engenharia Elétrica."

Por

Jean Marcos de Souza Ribeiro
Engenheiro Eletricista - FEIS / UNESP

Orientador: **Prof. Dr. José Paulo Fernandes Garcia**

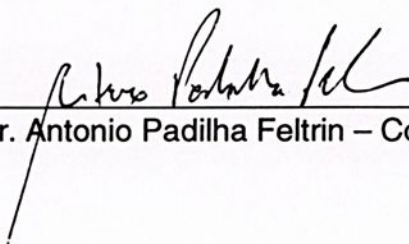
Ilha Solteira, Agosto de 2001

BCpIS - FEIS - UNESP

“Automação Industrial: Uma Proposta para Ensino no Curso de Engenharia Elétrica”

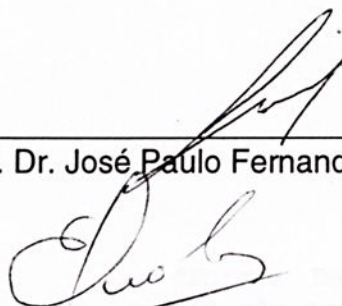
Jean Marcos de Souza Ribeiro

DISSERTAÇÃO SUBMETIDA À FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA – UNESP – COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA ELÉTRICA (ME).



Prof. Dr. Antonio Padilha Feltrin – Coordenador

COMISSÃO EXAMINADORA:



Prof. Dr. José Paulo Fernandes Garcia - orientador



Prof. Dr. Edwin Avolio



Prof. Dr. Luís Carlos Origa de Oliveira

Ilha Solteira – SP, agosto de 2001

"Porque Dele, por Ele e para
Ele são todas as coisas. Glória
pois, a Ele eternamente. Amém."

Romanos 11:36



AGRADECIMENTOS

A Deus, pelo seu amor, afeto, fidelidade e, principalmente, por sua imensurável graça e misericórdia, que constantemente se mostram em minha vida em forma de capacitação e sabedoria para transpor os momentos difíceis encontrados no caminho.

Aos meus pais, que não medem esforços para suprir cada uma de minhas necessidades, que me encorajam a seguir em frente e que sem os quais não seria tão motivado a buscar meus objetivos.

As minhas irmãs, que sempre me incentivaram com palavras de apoio, ajudando a alcançar meus objetivos.

Ao meu orientador Prof. José Paulo Ferrnandes Garcia, pela paciência, sabedoria, amizade, dedicação, respeito e, principalmente, por se cogar em formar bons pesquisadores, atacando junto nossos problemas e dificuldades.

Aos professores Marcelo Teixeira, Bevilatto Lizeis, Origa, Rossi e Edvaldo, que contribuíram imensamente na realização deste trabalho.

Aos técnicos do Laboratório de Engenharia Elétrica Adilson, Chavez, Everaldo, Aderson e Hidemasa, que auxiliaram durante a implantação do laboratório e sem os quais este trabalho não teria alcançado seus objetivos e, também, pela amizade e apreço cultivados ao longo desses anos.

Aos amigos, por serem, por inúmeras vezes, fonte de inspiração e de força para prosseguir. Ao amigo Marcos Chavez, que esteve sempre ao meu lado nos momentos difíceis de minha vida acadêmica e pessoal, aconselhando e dando apoio em minhas decisões:

A Roberta, pela plena compreensão de minhas ausências em alguns momentos de nossas vidas, pela amizade, carinho, amor, dedicação e pelo ânimo fornecidos nos momentos de fraqueza.

Aos pastores e membros da Igreja Apostólica, pelas orações e amizade.

Ao Departamento de Engenharia Elétrica e dedicação oferecidas aos seus pesquisadores.

E, finalmente, ao CNPq, pelo apoio financeiro, durante toda realização do trabalho de mestrado.

Aos meus pais que me ensinam os grandes valores da vida, que me dão amor, carinho e toda força para continuar, meus, pelo apoio e

Dedico



AGRADECIMENTOS

A Deus, pelo seu amor, afeto, fidelidade e, principalmente, por sua imensurável graça e misericórdia, que constantemente se mostram em minha vida em forma de capacitação e sabedoria para transpor os momentos difíceis encontrados no caminho.

Aos meus pais, que não medem esforços para suprir cada uma de minhas carências, que me encorajam a seguir em frente e que sem os quais não seria tão motivado a buscar meus objetivos.

Às minhas irmãs, que sempre me incentivaram com palavras de apoio, ajudando a alcançar meus objetivos.

Ao meu orientador Prof. José Paulo Fernandes Garcia, pela paciência, sabedoria, amizade, dedicação, respeito e, principalmente, por se engajar em formar bons pesquisadores, atacando junto nossos problemas e dificuldades.

Aos professores Marcelo Teixeira, Bovolato, Lizete, Origa, Rossi e Edvaldo, que contribuíram imensamente na realização deste trabalho.

Aos técnicos do Laboratório de Engenharia Elétrica Adílson, Chaves, Everaldo, Aderson e Hidemassa, que auxiliaram durante a implantação do laboratório e sem os quais este trabalho não teria alcançado seus objetivos e, também, pela amizade e apreço cultivados ao longo desses anos.

Aos amigos, por serem, por inúmeras vezes, fonte de inspiração e de força para prosseguir. Ao amigo Marcos Chaves, que esteve sempre ao meu lado nos momentos difíceis de minha vida acadêmica e pessoal, aconselhando e dando apoio em minhas decisões.

À Roberta, pela plena compreensão de minhas ausências em alguns momentos de nossas vidas, pela amizade, carinho, amor, dedicação e pelo ânimo fornecidos nos momentos de fraquezas.

Aos pastores e membros da Igreja Apostólica Ministério Comunidade Cristã pelas orações e amizade.

Ao Departamento de Engenharia Elétrica e seus funcionários, pelo apoio e dedicação oferecidos aos seus pesquisadores.

E, finalmente, ao CNPq, pelo apoio financeiro, durante toda realização do trabalho de mestrado.



RESUMO

Este trabalho apresenta uma proposta para implantação de um laboratório de ensino em automação de processos, para dar suporte a algumas disciplinas do curso de engenharia elétrica. Trata-se de um texto que incorpora as experiências obtidas durante a implementação do laboratório e a contribuição que este realizou na disciplina *Prática em Controle e Acionamentos de Máquinas Elétricas em Processos Industriais*, com enfoque eminentemente aplicativo, do curso de engenharia elétrica da Faculdade de Engenharia de Ilha Solteira (FEIS). Esta disciplina é fruto das pesquisas desenvolvidas ao longo dos últimos dois anos.

No texto enfoca-se a utilização do ambiente laboratorial para o ensino de automatização de processos, utilizando conhecimentos teóricos obtidos em outras disciplinas, regularmente oferecidas em curso de engenharia elétrica. Propõe-se a montagem de simulações de diversos processos, cuja automatização é conseguida utilizando-se como ferramenta principal os Controladores Lógicos Programáveis (CLPs).

Além de diversas experiências de laboratório, este trabalho também contribui ao ensino, contrapondo-se, através de simulação em laboratório, as duas formas de se automatizar um processo: i) método tradicional e ii) método tecnologicamente mais avançado. Desta maneira, é fornecida, aos alunos, uma maior motivação, proporcionada pelo fato de que eles têm a oportunidade de observar em funcionamento os processos automatizados, como decorrência dos conhecimentos adquiridos com a evolução da disciplina e do trabalho individual e em equipe, num ambiente laboratorial que retrata, por exemplo, os problemas que eventualmente terão numa indústria.



ABSTRACT

This work presents a proposal for implantation of a teaching laboratory in automation of processes, to give support the some disciplines of the course of electric engineering. It is a text that incorporates the experiences obtained during the implementation of the laboratory and the contribution that this accomplished in the discipline *Practice in Control and Connections of Electric Machines in Industrial Processes*, with focus eminently application, of the course of electric engineering of University of Engineering of Ilha Solteira (FEIS). This discipline is resulted of the researches developed along the last two years.

In the text the use of the laboratory environment is focused for the teaching of automation of processes, that uses theoretical knowledge obtained in other disciplines, offered in the course of electric engineering. Simulations of several processes is proposed, us which the automation is gotten being used as main tool the Programmable Logical Controllers (PLCs).

Besides several laboratory experiences, this work also contributes to the teaching, opposing, through simulation in laboratory, the two forms of automating a process: i) traditional method and ii) method technologically more advanced. This way, it is supplied to the students, a larger motivation, proportionate for the fact that they have the opportunity to observe in operation the automated processes, as consequence of the acquired knowledge with the evolution of the discipline and of the individual work and in team, in an laboratory environment that presents, for example, the problems that eventually will have in an industry.



ÍNDICE

CAPÍTULO 1 - INTRODUÇÃO	1
1.1 Motivação	3
1.2 Educação Profissional	3
1.3 Histórico da Automação Industrial.....	4
1.4 Aplicação Atual dos Controladores Programáveis.....	6
1.5 Sistema de Supervisão	7
1.6 Noções de Rede	11
1.6.1 Níveis em uma Rede Industrial.....	12
CAPÍTULO 2 - ESTADO DA ARTE.....	14
2.1 Automação no Setor Sucroalcooleiro.....	14
2.2 Automação na Indústria Automobilística.....	16
2.3 Automação na Indústria Alimentícia.....	17
2.4 Automação em Plataformas Offshore	18
2.5 Automação Industrial	18
2.6 Automação de Usinas Geradoras de Eletricidade	20
2.7 Automação Residencial	21
2.8 Racionalização Inteligente Utilizando CLP	24
CAPÍTULO 3 - ENSINO DA AUTOMAÇÃO EM CURSO DE ENGENHARIA ELÉTRICA	26
3.1 Ambiente Laboratorial.....	26
3.2 Metodologia de Ensino em Graduação.....	32
3.2.1 Metodologia para Primeira Etapa do Ensino	32
3.2.2 Metodologia para Segunda Etapa do Ensino	34
3.2.3 Método de Avaliação e Conceito Final.....	36
3.3 Proposta de Laboratório de Pesquisa em Automação	36
3.3.1 Justificativa	37
3.3.2 O Laboratório.....	37
CAPÍTULO 4 - SUGESTÕES DE EXPERIÊNCIAS	39
4.1 Experiência 1: Teste das E/S do CLP	39
4.2 Experiência 2: Lógica AND, OR, NAND, NOR.....	40
4.2.1 Lógica AND.....	40
4.2.2 Lógica OR.....	40
4.2.3 Lógica NAND.....	40
4.2.4 Lógica NOR.....	41
4.3 Experiência 3: Aplicação da Lógica para um Decodificador de Sete Segmentos.....	41
4.4 Experiência 4: Medida de velocidade retilínea	46
CAPÍTULO 5 - PROPOSTA FINAL: UMA INDÚSTRIA	49
5.1 Controlador de Nível.....	50
5.2 Misturador de Líquidos	54
5.3 Sistema de Refrigeração.....	58
5.4 Processos em Rede LAN	61



CAPÍTULO 6 - CONCLUSÃO E OBSERVAÇÕES FINAIS	64
CAPÍTULO 7 - BIBLIOGRAFIA	66
APÊNDICE A - SOFTWARE DE PROGRAMAÇÃO: LOGICMASTER TCP/IP	
V9.02.....	67
A.1 Apresentação do <i>Software</i>	67
A.2 Menu <i>Config</i>	69
A.3 Menu Program	74
A.4 Transferindo o programa e a configuração	78
A.5 Alguns blocos de programação	80
A.5.1 Bloco PID.....	80
A.5.2 Bloco COMM_REQ	83
A.6 Configuração dos CLPs	85
APÊNDICE B - DETALHES DOS PROCESSOS UTILIZADOS NA SIMULAÇÃO	
LABORATORIAL DA INDÚSTRIA	92
B.1 Processo de mistura e aquecimento de líquidos.....	92
B.1.1 Lógica de Controle Utilizando Diagrama Elétrico.....	93
B.1.2 Lógica de Controle Utilizando Diagrama Ladder.....	99
B.2 Controlador de nível de líquido.....	109
B.2.1 Lógica de Controle Utilizando Diagrama Elétrico.....	111
B.2.2 Lógica de Controle Utilizando Diagrama Ladder.....	114
B.3 Sistema de Refrigeração e sinalização via Pêndulo invertido.....	118
B.3.1 Lógica de Controle Utilizando Diagrama Elétrico.....	120
B.3.2 Lógica de Controle Utilizando Diagrama Ladder.....	124
B.3.3 Sistema Pêndulo Invertido	131
APÊNDICE C - CRIAÇÃO DAS TELAS DE SUPERVISÃO.....	141
C.1 Tela de Supervisão e Controle para o Processo de Mistura e Aquecimento de Líquido.....	150
C.1.1 Inserindo Objetos na Tela	150
C.1.2 Configurando os Objetos.....	156
APÊNDICE D - PROJETO COMPLETO DO CONTROLE E SUPERVISÃO DOS	
PROCESSOS UTILIZANDO O CIMPLICITY HMI.	189
D.1 Screens	189
D.2 Devices.....	189
D.3 Points <i>ID</i>	190
D.4 Ports	191
D.5 Events <i>ID</i> e Action <i>ID</i>	191
D.6 Scripts	192



CAPÍTULO 1

INTRODUÇÃO

A evolução tecnológica nas áreas de microeletrônica e informática tem permitido aumentar o grau de automação da produção industrial com melhor qualidade e menor custo, com desempenho e confiabilidade crescentes.

A necessidade de formar profissionais com uma boa visão de um processo automatizado e com a capacidade de utilizar adequadamente, neste processo, as novas tecnologias, levou ao trabalho de pesquisa em automação, que culminou na implantação de um laboratório de automação de processos.

Este trabalho tem por objetivo enfocar, promover e enriquecer o ensino em automação industrial, no curso de Engenharia Elétrica, através da utilização de equipamentos disponíveis em Laboratório, em especial, o Controlador Lógico Programável (CLP), também chamado de PLC (*Programmable Logic Control*), um marco histórico no que se refere ao estudo e implementação da automação.

É proposto neste trabalho um curso para ser ministrado em escolas de engenharia, baseados na experiência adquirida no decorrer dos últimos anos, em que, com a ajuda dos trabalhos de pesquisa de uma equipe de pesquisadores (entre eles, alunos de iniciação científica, mestrando e técnicos de laboratório), foi possível a implementação de um laboratório de automação de processos industriais e, conseqüentemente, da disciplina *Prática em Controle e Acionamentos de Máquinas Elétricas em Processos Industriais*, no curso de Engenharia Elétrica da Faculdade de Engenharia de Ilha Solteira (FEIS/UNESP).

Tal disciplina é ministrada desde 1998, tendo dado formação à aproximadamente 120 alunos. Esta disciplina confronta uma maneira convencional e uma maneira, que utiliza uma tecnologia mais avançada, em aplicações de controles de processos automatizados.



Este trabalho só foi possível devido às condições laboratoriais atualmente existentes na FEIS/UNESP. Tais condições foram obtidas através dos projetos de melhoria da qualidade do ensino em engenharia, ditos REENGE e PROIN, financiados, respectivamente, pelos órgãos de fomento FINEP e CAPES.

Conforme os trabalhos pesquisas evoluíam, com a aquisição de novos equipamentos e novos conhecimentos, eram introduzidos esses novos resultados na disciplina ligada diretamente ao laboratório. Observa-se então, que esta área de ensino deve ser subsidiada por um grupo de pesquisa, cujo objetivo deva ser à busca das novas tecnologias emergentes no mercado, para subsequente incorporação no ensino da graduação. Assim, serão formados engenheiros preparados para atuar de forma adequada nesta área, cujo desenvolvimento tecnológico é bastante acirrado.

A implantação do laboratório de automação de processos foi centrada na utilização do CLP pelo fato de que, segundo previsões, este equipamento ainda será utilizado nos próximos dez anos em nível mundial, e que no Brasil há previsão de uso pelos próximos 20 anos. Assim, este laboratório poderá ser utilizado e viável para as próximas gerações de estudantes.

Nos itens seguintes desde capítulo são apresentados alguns conceitos básicos do tema automação industrial. A seguir, no capítulo 2, é apresentado, de forma resumida, o quadro atual da automação nos diversos segmentos da indústria brasileira. No capítulo 3 descreve-se, em detalhe, uma proposta de implementação e utilização do laboratório de ensino de automação de processos para cursos de engenharia elétrica, incluindo a sugestão do laboratório de pesquisa em ensino de automação.

No capítulo 4 são detalhadas algumas sugestões para experiências básicas em laboratório. No capítulo 5 sugere-se uma série de experiências, que constituem a simulação de uma grande indústria e que dará oportunidade aos estudantes utilizarem conhecimentos mais avançados obtidos durante a realização da disciplina, vinculada ao laboratório de ensino em automação de processos. Finalmente, o capítulo 6 refere-se as conclusões e observações finais deste trabalho.

São apresentados, como apêndice, detalhes para ensino do *software* de programação do CLP (fabricante *GeFanuc*), do *software* supervisor *Cimlicity*, também da



GeFanuc, de forma que é possível transmitir aos alunos, através deste material, os conhecimentos básicos necessários sem a leitura dos extensos manuais. Também sob a forma de apêndice, apresenta-se em detalhes, as etapas dos processos sugeridos no capítulo 5, possibilitando ao leitor a implementação, em laboratório, das experiências propostas.

1.1 MOTIVAÇÃO

A motivação para a implantação de um laboratório de ensino de automação nos cursos de graduação e pós-graduação em engenharia elétrica se deve ao fato que os níveis de qualidade e produtividade, impostos pelo mercado atual, só são obtidos de forma viável e competitiva se empregadas técnicas de automação nos processos de transformação e fabricação de produtos, sendo, portanto, necessário que os futuros profissionais nesta área tenham uma boa formação e uma visão bastante real de um processo automatizado, que é a proposta central da implantação deste laboratório.

Outro fator que motiva o estudo em automação é a observação de uma mudança positiva da mentalidade do empresariado brasileiro, que se tornou menos resistente a realização de investimentos em automação, quando entendeu os benefícios que ela proporciona. Observa-se, no entanto, uma carência de profissionais com qualificação técnica suficiente para atender às exigências dessas empresas, fato justificado principalmente pela inexistência quase que total de laboratórios específicos, nesta área, nas escolas de engenharia do país.

Logo, este trabalho pretende contribuir não somente na formação do aluno, mas também incentivar a criação de novos cursos que utilizem os recursos tecnológicos já disponíveis no mercado, promovendo o gosto e o sentido da necessidade da inovação tecnológica, visto que as tecnologias de automação industrial vêm sendo, a cada dia, mais aperfeiçoadas, especialmente em termos de controle.

1.2 EDUCAÇÃO PROFISSIONAL

Os maiores desafios do mercado atual estão relacionados ao avanço tecnológico e as novas expectativas das empresas, que agora enfrentam mercados globalizados e extremamente competitivos.



Neste contexto, a educação profissional se tornou de fundamental importância na formação dos alunos de engenharia elétrica, sendo que o ensino deverá estar sempre atualizado e evoluindo paralelamente às necessidades do mercado de trabalho. Pensando desta forma, foi proposto e implementado um laboratório de ensino de automação de processos, para o curso de Engenharia Elétrica na FEIS, cuja utilização se divide em duas etapas, cada uma delas utilizando uma ferramenta para o controle de uma planta: i) automação industrial utilizando dispositivos eletromecânicos (relés, temporizadores...), e ii) automação industrial utilizando o CLP.

Ao apresentar essas duas maneiras de controlar uma planta industrial, a idéia é mostrar ao aluno as várias formas, incluindo aí as novas tendências tecnológicas no setor de automação industrial.

1.3 HISTÓRICO DA AUTOMAÇÃO INDUSTRIAL

O Controlador Lógico Programável seu surgimento na década de sessenta. A evolução tecnológica dos CLPs nos últimos 30 a, ou simplesmente CLP, tem revolucionado os sistemas de comando e controle industrial, desde nos é um dos mais nobres exemplos da aplicação da eletrônica nos processos de automação industrial. Antes do surgimento dos CLPs, as tarefas de comando e controle de máquinas e processos industriais eram feitas por relés eletromagnéticos, especialmente projetados para este fim.

Quando surgiram, na década de sessenta, os controladores programáveis tinham então a missão de substituir a lógica de relés, e isto ocorreu, pois a indústria automotiva, mais precisamente a *General Motors*, a cada novo lançamento ou modificação de um automóvel no mercado tinha que remodelar sua linha de montagem, sendo que esta remodelação significava a mudança da lógica dos painéis de relés eletromecânicos (painéis que chegavam a ter várias dezenas de metros de comprimento), o que levava muito tempo e por vezes no sucateamento e substituição do painel antigo.

A solução encontrada naquela época foi a construção de um “*hardware standard*”, que permitia sua utilização em situações diferentes, bastando para isso uma mudança na fiação. Na realidade, esse *hardware* era um amontoado de relés, sobre placas, com um determinado *layout*. Esta placa continha os terminais onde eram feitas as conexões



dos fios, daí dizer que o circuito de fiação era na realidade o *software* de programação. Isso permaneceu até a metade dos anos 70, quando os relés foram trocados por transistores e, desta forma, o “*hardware standard*” começou a ser composto por circuitos integrados.

Nascia assim, um equipamento bastante versátil e de fácil utilização, que vem se aprimorando constantemente, diversificando cada vez mais os setores industriais e suas aplicações.

Os Controladores Lógicos Programáveis (CLPs) surgiram então para executar lógicas de intertravamento digitais com várias vantagens. Dentre elas pode-se ressaltar a compactação dos painéis dada a reduzida dimensão dos CLPs e a flexibilidade da lógica de intertravamento, permitindo assim a reutilização do painel e a redução acentuada do tempo de remodelação da linha de produção.

Com o passar do tempo os controladores programáveis passaram a tratar variáveis analógicas e no início dos anos oitenta incorporaram a função do controle de malhas de instrumentação, com algoritmos de controle proporcionais, integrais e derivativos (PID). Ainda na década de oitenta, com a evolução dos microcomputadores e das redes de comunicação entre os CLPs, passaram a elevar seu desempenho, permitindo que vários controladores programáveis pudessem partilhar os dados em tempo real e que nesta mesma rede estivessem conectados vários microcomputadores, os quais, através de um *software* de supervisão e controle, podiam monitorar e comandar os processos como um todo, à partir de uma sala de controle, que não necessariamente precisava estar localizada no setor de produção.

Basicamente, um controlador programável apresenta as seguintes características:

- *Hardware* e/ou dispositivo de controle de fácil e rápida programação ou reprogramação, com a mínima interrupção da produção;
- Capacidade de operação em ambiente industrial;
- Sinalizadores de estado e módulo tipo *plug-in* de fácil manutenção e substituição;
- *Hardware* ocupando espaço reduzido e apresentando baixo consumo de energia;



- Possibilidade de monitoração do estado e operação do processo ou sistema, através da comunicação com computadores;
- Compatibilidade com diferentes tipos de sinais de entrada e saída;
- Capacidade de alimentar, de forma contínua ou chaveada, cargas que consomem correntes de até 2A;
- *Hardware* de controle que permite a expansão dos diversos tipos de módulos, de acordo com a necessidade;
- Custo de compra e instalação competitivo em relação aos sistemas de controle convencionais;
- Possibilidade de expansão da capacidade de memória, e
- Conexão com outros CLPs e dispositivos inteligentes (sensores, inversores, CNC, etc), através de rede de comunicação, podendo-se utilizar vários protocolos de comunicação.

1.4 APLICAÇÃO ATUAL DOS CONTROLADORES PROGRAMÁVEIS

A aplicação atual dos Controladores Lógicos Programáveis, não se restringe apenas ao controle dos dispositivos do “chão de fábrica”, através da realização de funções de intertravamento digital, mas também executam as funções de conectividade e aquisição de dados de todos os dispositivos, bem como funções de integração, concentração e envio dos dados do campo para o nível corporativo/gerência da empresa.

Vale ressaltar que a conectividade foi possível a partir do desenvolvimento, pelos fabricantes dos CLPs, de Módulos Dedicados para a comunicação com dispositivos de outros fabricantes, tais como inversores de frequência, sensores, balanças, conjunto de válvulas inteligentes, etc, e “Gateways” Universais que são módulos que emulam um computador PC/AT, os quais permitem por parte do usuário o desenvolvimento de protocolos de comunicação específicos para os dispositivos que se encontram no seu respectivo “chão de fábrica”.

Atualmente com a sedimentação do conceito das redes de campo “Field-Bus”, estas vêm se mostrando a melhor opção para a conectividade dos dispositivos do chão de



fábrica em função do seu custo/benefício ser razoável e da facilidade de sua aplicação, pois não existe a necessidade de desenvolvimento de protocolos, bastando apenas a simples parametrização dos dispositivos que conectarmos à rede.

Quanto a integração de CLPs, esta característica tornou-se viável a partir do momento em que as redes proprietárias entre controladores programáveis passaram a melhorar seu desempenho, chegando em torno de 1Mbps (hoje já existem protocolos com velocidade de 100Mbps, como por exemplo a versão "*Fast Ethernet*"), no entanto, a conexão de dispositivos de outros fabricantes ("*Multi-vendor*") não estava equacionada, sendo que para cobrir esta lacuna foram desenvolvidos *Gateways* de comunicação, os quais nada mais eram do que microcomputadores industriais, nos quais eram inseridas duas placas de comunicação, uma do fabricante do CLP a qual se conectaria a rede do CLP desse mesmo fabricante, e outra, a do fabricante do dispositivo estranho ao CLP que desejamos integrar, como por exemplo: inversores de frequência, Controladores Lógicos Programáveis de outros fabricantes, etc.

Para que a comunicação entre as placas ocorresse, era necessário desenvolver programas com rotinas em "*BASIC*" ou "*C*", o que tornava esta solução viável, no entanto de alto custo e complexidade.

Para esta função de integração, cresce hoje no mercado a utilização de redes não proprietária, ou seja, abertas, cujo padrão elétrico e protocolo de comunicação são de conhecimento público. Neste caso, as redes *Ethernet* passaram a se definir como um padrão no setor industrial. Sua popularidade e sua elevada velocidade de transmissão permitiram a adoção desta solução nas aplicações industriais, no entanto, ela exige do controlador programável uma elevada capacidade de processamento e de uma alta disponibilidade de massa de memória para viabilizar a comunicação em "tempo real".

1.5 SISTEMA DE SUPERVISÃO

Os Sistemas Supervisórios podem ser vistos como sistemas que supervisionam ou monitoram processos executados em uma planta industrial, através da visualização de variáveis da planta que está sendo automatizada, bem como das ações tomadas pelo sistema de automação.



O *software* de supervisão, localizado no nível de controle do processo das redes de comunicação, é o responsável pela aquisição e envio de dados, entre os controladores lógicos programáveis e o computador, pela sua organização, utilização e gerenciamento dos dados. Poderá ser configurado para taxas de varredura diferentes entre CLPs e inclusive, entre pontos de um mesmo CLP.

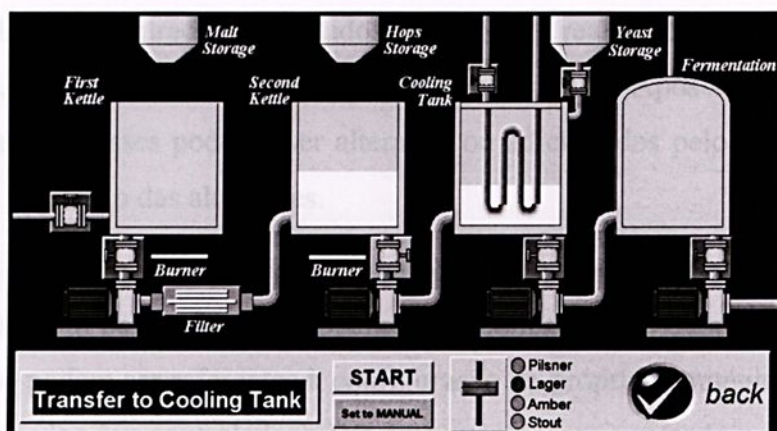


Figura 1.1 - tela ilustrativa de um software de supervisão e controle.

Os dados da planta controlada, obtidos através do software de supervisão, devem ser condicionados e convertidos em unidades de engenharia adequadas, em formato simples ou de ponto flutuante, armazenando-os em um banco de dados operacional. A configuração individual de cada ponto, supervisionado ou controlado, permite ao usuário definir limites para alarmes, condições e textos para cada estado diferente de um ponto, valores para conversão em unidade de engenharia, etc.

O *software* deve permitir que estratégias de controle possam ser desenvolvidas utilizando-se de funções avançadas, através de módulos dedicados para implementação de funções matemáticas e *booleanas*, por exemplo. Através destes módulos poderá ser feito, no *software* aplicativo de supervisão, o controle das funções do processo.

Os dados obtidos podem ser manipulados de modo a gerar valores para parâmetros de controle como *set-point*. Os dados são armazenados em arquivos de dados padronizados. Estes arquivos poderão ser acessados por programas de usuários para realização de cálculos, alteração de parâmetros e dos seus próprios valores.

O *software* supervisor é visto como o conjunto de programas gerado e configurado no *software* básico de supervisão, implementando as estratégias de controle e

supervisão, as telas gráficas de interfaceamento homem-máquina (IHM), a aquisição e tratamento de dados do processo, a gerência de relatórios e alarmes. Este *software* deve ter entrada de dados manual, através de teclado.

Os dados serão requisitados através de telas com campos pré-formatados que o operador deverá preencher. Estes dados deverão ser auto-explicativos e possuírem limites para as faixas válidas. A entrada dos dados deve ser realizada por telas individuais, sequencialmente, com seleção automática da próxima entrada. Após todos os dados de um grupo serem inseridos, esses poderão ser alterados ou adicionados pelo operador, que será o responsável pela validação das alterações.

A estratégia de supervisão e controle é desenvolvida com o *software* básico de supervisão que cria um banco de dados operacional com todos os dados de configuração do sistema. Os dados podem ser referentes à configuração da própria estratégia ou referentes aos pontos supervisionados (ou controlados). Em ambos os casos, o método e recursos utilizados para entrada de dados devem ser compostos por ferramentas para *Windows*.

O conjunto de telas do *software* de supervisão deve permitir os operadores, controlar e supervisionar completamente toda a planta. As telas deverão ser organizadas em estrutura hierárquica do tipo árvore, permitindo um acesso sequencial e rápido. A seguir, são descritas as principais telas que o aplicativo deve conter:

- **Telas de visão geral:** são telas que apresentam ao operador uma visão global de um processo, sob visualização imediata na operação da planta. Nestas telas são apresentados os dados mais relevantes à operação e objetos que representam o processo. Os objetos devem ser dotados de características dinâmicas, representando o estado de grupos de equipamentos e áreas do processo apresentado. Os dados devem procurar resumir, de forma significativa, os principais parâmetros a serem controlados (ou monitorados) do processo específico;
- **Telas de grupo:** são telas representativas de cada processo ou unidade, apresentando objetos e dados de uma determinada área. Os objetos devem ser dotados de características dinâmicas representando o estado e/ou condições dos equipamentos da área apresentada. Os dados apresentados devem representar valores quantitativos dos parâmetros supervisionados (ou controlados). As telas de grupo também possibilitam ao operador, acionar os equipamentos da área através de comandos do tipo abrir/fechar ou ligar/desligar. Além disso,



o operador poderá alterar os parâmetros de controle ou supervisão, tais como *set-point*, limites de alarme, modos de controle, etc.;

- **Telas de detalhe:** são telas que atendem a pontos e equipamentos controlados (ou monitorados) individualmente. Serão compostas, quando possível, por objetos com características dinâmicas, representando o estado do equipamento. Os dados apresentam todos os parâmetros do ponto supervisionado. As telas devem possibilitar ao operador alterar os parâmetros do equipamento, seus limites, os seus dados de configuração, etc., e

- **Telas de manutenção:** são compostas por informações de problemas, alarmes, defeitos e dados de manutenção das diversas áreas referentes ao processo e equipamentos destes, incluindo o próprio sistema de controle. As informações são do tipo histórico de falhas, programa de manutenção dos equipamentos (corretiva e preventiva), e informações gerais dos equipamentos (comerciais, assistências técnica, etc.). O histórico de falhas por equipamento ou área fica armazenado em arquivos no banco de dados do *software* de supervisão, possibilitando o tratamento destas informações através de telas, que servirão de orientação à manutenção, ou através de programas de usuário para estatísticas de utilização e defeitos.

O *software* básico de supervisão possui um módulo para desenvolvimento de relatórios. Criados em formato padrão, para relatórios do tipo históricos, permitem ao operador a escolha de quais variáveis deseja visualizar. Os dados podem ser apresentados nas telas das estações com campos de identificação para “TAG”, data, hora e descrição do ponto.

Os relatórios poderão ser solicitados manualmente pelo operador e destinados para impressoras ou terminais de vídeo. Os dados históricos são armazenados em arquivos de modo que podem ser acessados pelos programas de relatórios, para serem trabalhados e apresentados à operação. Deste modo, os arquivos podem ser armazenados em meios magnéticos para utilização futura.

O laboratório de ensino em automação de processos da FEIS dispõe do *Cimplicity-HMI*, que é um *software* para criação de aplicativos de supervisão e controle de processos nas mais diversas áreas, tais como: química, automação predial, manufatura, elétrica, segurança, laboratórios de testes, saneamento e máquinas. Desenvolvido em um ambiente avançado, orientado a objetos e de fácil configuração. Permite a visualização e



acionamento de variáveis, bancos de dados, relatórios, receitas e conectividade com todo tipo de sistema, inclusive via Internet.

Este aplicativo oferece sofisticados recursos que otimizam o tempo de desenvolvimento e a manutenção dos sistemas:

- Interface clara, lógica e intuitiva;
- Conectividade com a maioria dos equipamentos disponíveis no mercado ou mesmo com outros aplicativos *Windows*;
- Biblioteca gráfica para criação de telas;
- Suporte à rede e arquitetura cliente/servidor;
- Configuração e reconhecimento de alarmes;
- Relatórios formatados, graficamente customizados pelo usuário;
- Registro de dados em disco e análise histórica;
- Receitas que permitem a programação de valores para o envio ao processo;
- *Scripts* que permitem a criação de rotinas exclusivas, definindo lógicas e criando seqüências de atitudes através de uma linguagem de programação interativa, personalizando ao máximo o aplicativo;
- Suporte a banco de dados via *ODBC (Open Data Base Connectivity)* - *Access, SQL Server, Oracle, dBase*;
- Módulo matemático para a formulação de equações;
- Controle de acesso por nível de usuário;
- Acesso remoto via Internet, e
- Aquisição de eventos com precisão de 100ms.

1.6 NOÇÕES DE REDE

Diante do avanço tecnológico e da globalização, que nos permite desfrutar de muitos benefícios, apresenta-se no mercado industrial um novo cenário, onde os interesses



deixaram de ser os produtos e toda a atenção ficou voltada para as novas tecnologias incorporadas às redes industriais *ControlNet*, *DeviceNet*, *Fieldbus* e *Ethernet*. As redes, na atualidade, são indispensáveis na Automação pelas facilidades que oferecem na comunicação e em função do domínio que se possui sobre elas.

Em uma empresa existem uma série de equipamentos e dispositivos dedicados ao controle de uma máquina ou de uma parte fechada do processo. Entre os dispositivos estão os CLPs, que devido ao avanço tecnológico tem um papel importante numa rede industrial de comunicação.

As redes industriais têm estabelecido uma forma de unir todos os dispositivos, aumentando o rendimento e proporcionando novas alternativas de interoperabilidade. As vantagens de utilizar uma rede de comunicação para controlar determinado processo industrial são, entre outras as seguintes:

- Visualização e supervisão de todo o processo produtivo;
- Tomada de dados do processo de forma mais rápida e instantânea;
- Melhoria do rendimento geral de todo o processo;
- Possibilidade de intercâmbio de dados entre setores do processo e entre departamentos, e
- Programação à distância, sem necessidade de estar na fábrica.

As vantagens são evidentes, mas a mudança trará um certo custo que deve ser estudado para determinar se o investimento é rentável ou necessário.

1.6.1 Níveis em uma Rede Industrial

Em uma rede industrial coexistem equipamentos e dispositivos de todo tipo, os quais são agrupados hierarquicamente para estabelecer conexões mais adequadas de cada área. Desta forma, se definem quatro níveis dentro de uma rede industrial.

- **Nível de gestão:** é o nível mais elevado, que se encarrega de integrar os níveis seguintes em uma estrutura de fábrica, e incluir múltiplos gerenciamentos. As máquinas aqui conectadas serão estações de trabalho que fazem uma ponte entre o processo produtivo e a área de gestão,



na qual são supervisionados as vendas e estoques. Emprega-se uma rede do tipo *LAN* (*Local Area Networks*) ou *WAN* (*Wide Area Network*).

- **Nível de controle:** é encarregado de conectar e dirigir as distintas zonas de trabalho. A este nível se situam os robôs de alta linha e os computadores para projeto, controle de qualidade, programação, etc. Pode-se empregar uma rede tipo *LAN*.
- **Nível de campo e processo:** é encarregado de uma integração de pequenos mecanismos (computadores compactos, multiplexadores de E/S, controladores PID, etc.) dentro de sub-redes. No nível mais alto destas redes, pode-se encontrar um ou vários computadores modulares, atuando como mestres da rede ou mestres flutuantes. Neste nível se empregam os barramentos de campo.
- **Nível de E/S:** é o nível mais próximo de todo processo. Aqui estão os sensores e atuadores, encarregados de controlar os recursos do processo produtivo e tomar medidas necessárias para correta automação e supervisão.

Apresenta-se no próximo capítulo um breve relato dos principais setores onde a automação se faz necessário, e como estão sendo empregados os dispositivos de controle de processos, com tecnologias mais avançadas.

Em todos os níveis de automação citados acima o CLP é uma ferramenta fundamental para sua realização, podendo ser usado para atuar diretamente no processo, ou como um banco de dados a ser manipulado por um outro controlador. Por isso, todos aqueles que desejam trabalhar em automação terão que ter o conhecimento deste equipamento.

Nos itens a seguir serão apresentados alguns setores onde a automação se apresenta como solução ou uma forma de se obter um produto de melhor qualidade e menor custo.

2.1 AUTOMAÇÃO NO SETOR SUCROALCOOLEIRO

Neste setor existe muito a ser pesquisado, pois no Brasil ainda existem várias unidades produtoras de álcool e açúcar, que não tem o básico em automação. Em contrapartida



ESTADO DA ARTE

Um dos segmentos que mais cresce no mundo é a automação e, portanto, também é um grande alvo para pesquisadores, entre eles, engenheiros eletricitas, mecânicos, de produção, químico, enfim, de uma gama enorme que tem o interesse direto ou indireto do resultado da pesquisa. A automação é subdividida em vários ramos, cada um tratando de um objeto diferente de controle, por exemplo, temos a automação residencial, industrial, petroquímica, sucroalcooleiro, de sistemas elétricos, no setor de papel e celulose, da distribuição de energia elétrica, de hidrelétricas (que é dividida novamente em vários ramos), entre muitas outras.

A necessidade de se automatizar um processo, veio da exigência que o mercado globalizado mundial, está impondo aos seus produtos. Para um produto nacional entrar no mercado com condições de competir com produtos de outro país, ou de outro fabricante, ele deve ter boa qualidade com um preço acessível, daí a automação existe como principal meio de se obter este produto de forma mais eficiente, barata e com possibilidades de evolução.

Em todos os ramos de automação citados acima o CLP é uma ferramenta fundamental para sua realização, podendo ser usado para atuar diretamente no processo, ou como um banco de dados a ser manipulado por um outro controlador. Por isso, todos aqueles que almejam trabalhar em automação terão que ter o conhecimento deste equipamento.

Nos itens a seguir serão apresentados alguns setores onde a automação se apresenta como solução ou uma forma de se obter um produto de melhor qualidade e menor custo.

2.1 AUTOMAÇÃO NO SETOR SUCROALCOOLEIRO

Neste setor existe muito a ser pesquisado, pois no Brasil ainda existem várias unidades produtoras de álcool e açúcar, que não tem o básico em automação. Em contrapartida



existem unidades que já planeja o uso de um controle avançado e otimização do processo utilizando novas tecnologias no processo de automação.

De maneira geral, este setor possui parte de suas unidades automatizadas, porém sem comunicação e não integradas a um sistema central.



Figura 2.1 - empresa sucroalcooleira.

Um pouco diferente das demais indústrias, onde a automação já é um item agregado aos projetos, o setor sucroalcooleiro, em geral, figura como um setor que está conhecendo a pouco tempo o que é a automação e os benefícios que ela pode proporcionar para o processo e, conseqüentemente, aos seus produtos.

O resultado deste atraso, nos investimentos em automação, não foi tão maléfico para as indústrias sucroalcooleiras do país; isto porque elas conseguiram evitar o que aconteceu com vários outros setores abrangidos pela automação, onde as empresas se viram totalmente dependentes de seus fornecedores, devido à aquisição de equipamentos proprietários (de baixa tecnologia, incapaz de comunicar e de interligar com equipamentos de diferentes fabricantes), utilizados para automação de seus parques industriais.

Atualmente, devido à padronização e abertura dos sistemas, pode-se conectar equipamentos de automação de diferentes fabricantes num mesmo sistema de controle, aumentando a flexibilidade e possibilidades de implementação de diversas novas tecnologias, de acordo com sua inserção no mercado.

2.2 AUTOMAÇÃO NA INDÚSTRIA AUTOMOBILÍSTICA

Já foi observado que o CLP surgiu graças às necessidades da indústria automobilística, aliás, a primeira aplicação de robô na manufatura também deu-se em uma montadora.

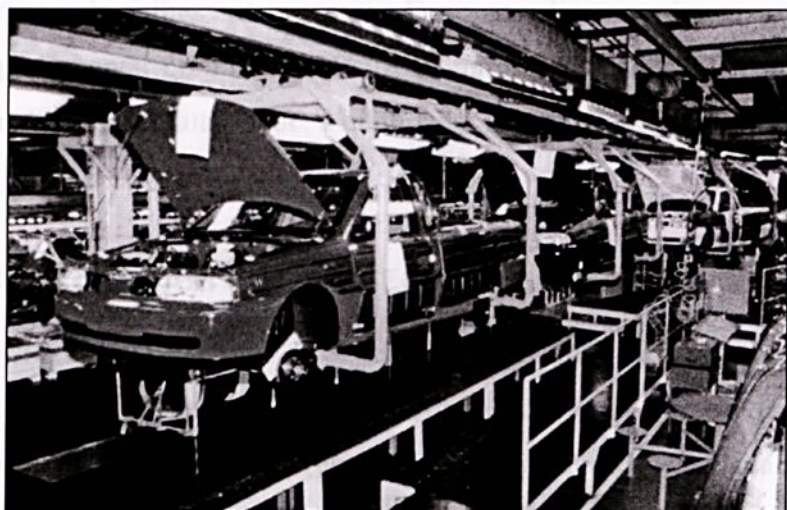


Figura 2.2 - indústria automobilística.

Esse segmento continua sendo precursor de novas tecnologias e tendências no que tange à automação como instrumento para elevar a produtividade, o padrão de excelência, refletindo na redução de custos.

Atualmente, as montadoras agregam tecnologia de ponta em todo ciclo produtivo, automatizando, por exemplo, equipamentos de solda, armação de carroceria, estamparia, usinagem, pintura, montagem, agregado motriz, equipamentos de teste, colocação de vidros, transportadores, enfim, a automação está em praticamente todo o processo.

Devido à potencialidade do mercado brasileiro e pela alta competição imposta pelo processo de globalização, a indústria automobilística brasileira prioriza investimentos para a modernização de suas plantas e o aprimoramento dos seus produtos.

Outro detalhe que não pode ser desconsiderado: com a abertura das fronteiras aos carros importados, o consumidor brasileiro passou a exigir mais em termos de qualidade, conforto e segurança dos veículos. Esta seletividade dos usuários impôs às montadoras, estabelecidas no país, o grande desafio de criar linhas fabris flexíveis para melhorar seus

processos e produtos. Assim, novas opções de modelos, *design* e cores estão sendo oferecidas com intervalos de lançamentos cada vez menores.

Independente da marca ou capacidade produtiva, um ponto comum entre as montadoras de veículos é que estas estão seguindo a lógica de diminuir custos e trabalhar com as menores margens de falhas no processo. E no atendimento destas exigências, o setor automobilístico pode contar com os benefícios da automação industrial como um importante instrumento para elevar a produtividade, a qualidade, garantir segurança da produção e reduzir custos.

2.3 AUTOMAÇÃO NA INDÚSTRIA ALIMENTÍCIA

Assim como os setores de papel e celulose, química e petroquímica, a indústria alimentícia adquiriu equipamentos que utilizavam tecnologias proprietárias, dependendo totalmente do fornecedor para obter qualquer tipo de melhoria no desempenho da automação. Muitos utilizavam o SDCCD (Sistema Digital de Controle Distribuído).



Figura 2.3 - indústria alimentícia.

Logo que o mercado começou a exigir maior eficiência na obtenção do produto, começaram a substituir a tecnologia existente por equipamentos aptos a fornecer as exigências do mercado, com flexibilidade e usando uma tecnologia de comunicação aberta, deixando toda dependência a determinados fabricantes. O equipamento utilizado para isso foi o CLP, com a utilização de *software* supervisor.

2.4 AUTOMAÇÃO EM PLATAFORMAS OFFSHORE

A construção de plataformas *offshore* da Petrobrás de 1982 a 1985 possuía partida com comandos elétrico-pneumáticos e salas de controle com alarmes simples e registros em papel. A evolução incorporou a eletrônica e os CLPs, nas construções das novas plataformas. Hoje, muitas já possuem uma estação central de operação e supervisão.

Como todos os setores em uma plataforma são considerados críticos, cada CLP possui um sobressalente - estratégia esta chamada de sistemas redundantes - garantindo o funcionamento interrompido do sistema automatizado no caso de falha em algum CLP.

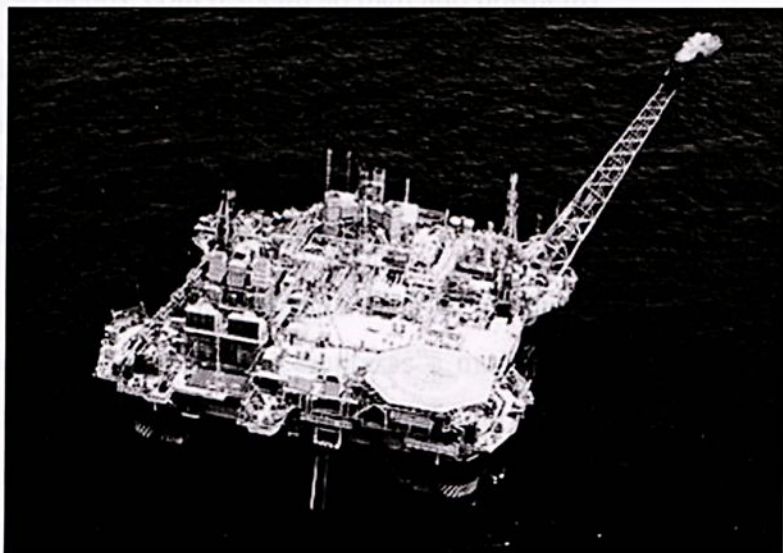


Figura 2.4 - plataforma offshore.

Mesmo com essa tendência tecnológica a ser aplicada a todas as plataformas *offshore*, existe uma grande dificuldade de se modificar o sistema em operação, visto que poderia comprometer toda segurança já estabelecida; sendo, portanto, necessário um profundo estudo das reais necessidades e os seus benefícios, na implantação destas novas tecnologias de controle.

2.5 AUTOMAÇÃO INDUSTRIAL

A indústria manufatureira vem ocupando uma posição muito importante na economia brasileira, sendo responsável por aproximadamente 40% do Produto Interno Bruto (PIB) e com uma participação de 30% nas exportações. A busca de uma maior integração e participação no mercado mundial têm sido, nos últimos anos, o objetivo da indústria nacional.

Desta forma, a indústria brasileira tem procurado otimizar todas as atividades relacionadas direta ou indiretamente à fabricação do produto, tais como projeto, planejamento e processos de fabricação, controle de qualidade, armazenagem e distribuição. Uma das ferramentas para se alcançar essa otimização é a utilização do CLP como dispositivo de supervisão, controle e transmissão de dados.

É inegável dizer que o grande avanço tecnológico com respeito à automação industrial é impulsionado, principalmente, devido à rápida evolução que vem ocorrendo nas áreas de computação e comunicação. Nota-se que o ritmo deste avanço não mostra sinais de desaceleração, pelo menos, com respeito ao mercado brasileiro.

Não se pode mais falar de tendências de utilização de dispositivos microprocessados, trabalhando com comunicação em rede, para realização de determinada tarefa, visto que hoje isso já é algo totalmente exequível, muito aplicado em automação industrial e até mesmo em automação residencial.

A situação atual do mercado mundial, no que se refere à automação industrial, indica forte crescimento, onde várias empresas e muitos fabricantes lutam por seu espaço, acirrando assim uma competição, que tem contribuído para o avanço tecnológico industrial. No Brasil essa disputa pelo mercado ainda não é tão acirrada, visto que o mercado brasileiro é, ainda, muito amplo para aplicações de automação. Mas aos poucos notamos que cada fabricante, de produtos voltados à automação, está adquirindo seu espaço.

Um fato muito importante, que impede, ou pelo menos minimiza, o domínio de um fabricante sobre os demais é a criação de uma arquitetura aberta de comunicação entre diferentes equipamentos de diferentes fabricantes. Logo, tecnologias proprietária de determinados fabricantes estão saindo do mercado, dando lugar a arquiteturas abertas, estreitando assim a margem que diferenciava um dispositivo de outro.

Diversas entidades internacionais preocupam-se com a criação e o aperfeiçoamento de normas específicas sobre CLPs. Algumas normas que merecem atenção são listadas a seguir:



- *NEMA (National Electrical Manufacturers Association) ICS 1-109 "tests and test procedures"*: define uma série de testes aplicáveis ao nível de projeto, produção e aplicação de CLPs;
- *NEMA ICS 2-230 "Components for solid-state logic systems"*: descreve testes e equipamentos para verificação de imunidade a ruído;
- *IEEE std 518-1977 "Guide for installation of electrical equipment to minimize electrical noise inputs to controllers from external sources"*: contém um estudo sistemático sobre ruídos, suas fontes e métodos de redução;
- *NEMA ICS 3-304 e IEC 1131*: são as normas mais completas sobre CLPs, abordando problemas de confiabilidade, redundância, recuperação de falhas, sistemas de comunicação, entre outros. Elas enumeram estruturas, tipos de programação e subsistemas para CLPs, estabelecem parâmetros máximos e mínimos de operação, dentro dos quais os CLPs devem funcionar sem apresentar defeitos, e descrevem testes padronizados para caracterizar e verificar as especificações dos fabricantes. A norma *IEC 1131* padroniza 5 linguagens para CLP: *IL*, *LD*, *GRAFCET*, *ST (Structured Text, semelhante a C)* e *FBD (Functional Block Diagram*, que transforma um desenho de um diagrama de blocos com portas *AND*, *OR*, *XOR*, etc., da lógica digital em programa de CLP).

2.6 AUTOMAÇÃO DE USINAS GERADORAS DE ELETRICIDADE

A primeira usina de grande porte que foi equipada com sistema de supervisão e controle foi a de Xingo. Nesta, além do sistema digital apresentar elevado grau de redundância, um sistema convencional foi instalado junto às unidades geradoras a fim de fornecer um controle local. O sistema digital instalado em Xingo compreende três subsistemas: Interface Homem Máquina (IHM); Interface com o Processo, e Comunicação e sincronização.

Depois do sucesso de Xingo, outras usinas também receberam equipamentos de supervisão e controle digital, destacando-se Porto Primavera, Tucuruí e Salto de Caxias.

Assim como em usinas, outro setor em que a automação está sendo empregada é no controle e monitoramento remoto das subestações. É o caso da Elektro que tem todas as



suas 83 subestações controladas remotamente a partir de sete Centros de Operação da Distribuição (CODs). Mas nem todas as companhias adotaram tal procedimento, por exemplo, a Eletropaulo Metropolitana possui apenas 20 subestações com algum tipo de automação, de um total de 125 subestações.

Na CPFL cerca de 40% das suas subestações já estão equipadas com sistema de supervisão e controle remoto. A CEMIG e a COPEL também estão bastante avançadas neste aspecto.

Entretanto, observa-se que a adoção da tecnologia está consolidada, já que todas subestações novas ou em processo de aquisição já estão vindo com sistema de monitoração, controle e proteção digital incorporados.

2.7 AUTOMAÇÃO RESIDENCIAL

Automação residencial é um sistema inteligente que integra, iluminação, entretenimento, segurança, telecomunicações, aquecimento, ar condicionado e muito mais; tudo isso controlado através de um sistema programável e centralizado.

No que se refere à automação residencial, muito se tem feito para fornecer maior conforto ao homem. Tanto a automação industrial como a residencial sempre visou o bem-estar do homem: uma no ambiente de trabalho, outra em seu lar.

As maiores diferenças que existem entre a automação residencial e automação industrial é a potência dos equipamentos e principalmente o ambiente onde irá trabalhar o controlador. Uma residência é menos sujeita a interferências do tipo eletromagnéticas e ruídos, por isso neste tipo de automação pode-se utilizar todo o potencial de comunicação e controle sem muitas restrições, desta forma métodos de controle muito interessantes podem ser implementados.

O primeiro benefício de um sistema de automação residencial é viabilizar a comunicação entre aparelhos individuais com o sistema de automação centralizado, cuidando de tarefas repetitivas. Por exemplo, ao sair de casa, o sistema de segurança poderá ser armado usando apenas um comando "Fora de Casa", e como resultado, o sistema de automação: 1) desligará lâmpadas desnecessárias; 2) colocará o termostato do ar condicionado em um valor que diminua o consumo de energia elétrica ou mesmo desligá-lo; 3) desligará o aparelho de



som e TV; 4) ativará a secretária eletrônica; 5) inicializará uma sequência programada de acender e apagar luzes, ligar e desligar o sistema de som e TV, etc. Tudo controlado pelo sistema de automação, simulando a presença de pessoas dentro da residência.

O melhor é que a programação dessa sequência poderá ser alterada à vontade pelo próprio usuário. Quando os moradores retornarem, o sistema de automação poderá recebê-los acendendo lâmpadas apropriadas, colocando música, ajustando a temperatura do ar condicionado, tudo isso poderá ser programado instantes antes dos moradores chegarem, simplesmente através de uma chamada telefônica utilizando um código de acesso, por exemplo.

Pode-se obter um maior ou menor grau de automação em sua residência como desejar. Se a residência for completamente automatizada, ao toque de um simples botão, um sistema de automação residencial aumentará o seu conforto melhorando seu estilo de vida, além de melhorar a praticidade e o conforto do seu dia a dia economizando tempo e dinheiro.

Outro exemplo vem do advento da comunicação *Wireless* (sem fio), que permite fazer com que todos os equipamentos da residência se comuniquem para efetuar certas tarefas: ligar uma máquina, abrir a cortina, acender uma luz, ligar o som, etc. Esse padrão de comunicação, que levou o nome de *Bluetooth*, já saiu do papel e alguns fabricantes de equipamentos já o adotaram; sua especificação é definida de curto alcance de transmissão e recepção, sendo de no máximo 100 metros para os dispositivos mais avançados, que utiliza um *link* de rádio entre 2.4 e 2.48 GHz. O maior objetivo deste tipo de comunicação é a interoperabilidade entre dispositivos de se comunicarem utilizando a mesma linguagem.

Embora já seja uma realidade o uso da comunicação *Wireless* para automação residencial, ainda há muito a se estudar para a utilização desta tecnologia em ambiente industrial, visto que neste os ruídos existentes em ambientes industriais causariam má operação destes dispositivos.

A automação residencial se tornou um ramo importante no que se refere à automação. A publicação de Leonardo Coutinho, na revista veja, em 9 de agosto de 2000, da matéria "Só falta falar" no item tecnologia, retrata bem os avanços alcançados neste ramo da automação.



“Uma construtora paulista está erguendo um edifício residencial de fazer inveja aos Jetsons – a família do futuro dos desenhos animados. O prédio fica num bairro de classe média alta de Santo André, no ABC paulista, será concluído no fim do ano e seus vinte apartamentos, quase todos já vendidos, são dotados de um moderníssimo sistema de automação. No Champs Elysées, como foi batizado o edifício, não será necessário usar interruptores de luz para ir de um cômodo a outro. As lâmpadas acenderão e se apagarão à medida que o morador for passando. Tempo quente? O ar-condicionado ligará automaticamente quando a temperatura se elevar muito. Janelas se fecharão sozinhas sempre que o vento ou a chuva aumentarem. Na suíte principal, um alto-falante informará em voz alta a presença de pessoas em qualquer cômodo. Ideal para controlar filhos insones transitando pela casa durante a madrugada. O cérebro do sistema é um computador instalado na sala e programado para servir o morador como ele quiser. É possível comandá-lo por meio de celular, da internet ou de controle remoto...” (9 de agosto de 2000 - Revista Veja)

Não foi mencionado qual a forma de automação residencial adotada no edifício *Champs Elysées*, mas generalizando este caso como um caso de automação residencial qualquer e estudando os objetivos de controle desta planta, observa-se que os equipamentos podem ser controlados utilizando sensores e atuadores, comunicando com um CLP; cada residência (apartamento) com uma rede de dispositivos, que utilizam determinado protocolo, tal como o TCP/IP, mandando informações para o CLP que tomará as decisões e reenviará um sinal para atuar no sistema ou para acionar um atuador.

Pode-se ligar o CLP a um sistema supervisório, ou seja, a uma plataforma PC que, além de monitorar e controlar todos os pontos do sistema poderá enviar sinal visual ou até mesmo sonoro, conforme descrito no texto anterior. Mais do que isto, estes pontos poderão estar à disposição do dono da residência via *Web*, podendo este atuar no sistema remotamente ou visualizar o estado dos pontos controlados/monitorados. O único inconveniente neste tipo de serviço remoto é que o PC deveria estar ligado permanentemente para disponibilizar os pontos na *Web*.

Navegando por sites de busca na *Web*, observa-se que existem várias empresas se dedicando a esse ramo da automação, ou então migrando parte da empresa de automação industrial para essa nova divisão, chamada “*Home Automation*”.



2.8 RACIONALIZAÇÃO INTELIGENTE UTILIZANDO CLP

Até o momento foram apresentados alguns setores onde a automação se apresenta como solução. Exposta a crise energética enfrentada no Brasil e também em alguns países, como nos EUA, é importante dizer que um processo automatizado é significativamente mais econômico, do ponto de vista energético, que um processo onde a intervenção do homem é utilizada. Basta dizer que num sistema automatizado uma máquina se comunica com outra e os sensores informam as situações momentâneas de vários pontos do processo, logo, o comando de ligar e desligar só irá ocorrer quando houver real necessidade, com precisão de poucos milissegundos.

Um exemplo bastante simples de como se pode comparar um processo automatizado com um processo controlado pelo homem é o de transferência de um determinado líquido, dentro de um reservatório, através de uma bomba de fluxo. No processo automatizado, é feita uma programação tal que a bomba desliga assim que o sensor de nível do reservatório indicar nível baixo, enquanto que com a intervenção humana pode-se haver um atraso, fazendo a bomba operar à vazio durante certo tempo.

O que se nota então, é que num sistema automatizado consegue-se um elevado nível de sincronismo entre sensores, atuadores e máquinas, evitando assim o dispêndio de energia elétrica. O CLP, por exemplo, é um dispositivo que trabalha com precisão na casa dos milissegundos, e que possui baixíssima potência, logo sua utilização num sistema automatizado tende a tornar o processo mais econômico. Outro fator que mostra a economia de energia com a utilização dos CLPs é que os processos ficam mais velozes, fazendo com que as máquinas operem durante um tempo menor para obter o mesmo resultado quando não se utiliza automação.

Os CLPs também podem ser utilizados em residência para controlar o consumo de energia elétrica, por exemplo, numa automação residencial utilizando CLP, pode-se programá-lo para que a luz da varanda só ligue depois das 19:00hs, limitar o tempo de chuveiro ligado, desligar a luz de um ambiente quando não existirem pessoas no local, desligar sistemas de refrigeração e aquecimento quando alcançada a temperatura desejada, desligar aparelhos eletrônicos, etc.



Algo muito interessante que pode ser feito com um CLP de pequeno porte, por exemplo, é o chaveamento da alimentação de uma residência para um sistema alternativo de energia solar, toda vez que faltar energia, ou durante algum tempo da noite. Observa-se então que, além das grandes vantagens de flexibilidade e eficiência na utilização dos CLPs, eles também se mostram como um meio de se economizar energia diante da crise de energia em que o país se encontra.

Como descrito no capítulo anterior, nota-se que a utilização do CLP em vários setores é uma maneira eficiente para automatizar um processo. Embora ele possa ser utilizado em várias aplicações, o setor onde ele causou maior impacto e alcançou maior sucesso, foi na indústria manufatureira, que o adota como a melhor ferramenta de controle de processo. Por ser capaz de trabalhar com variáveis analógicas o CLP começou a ser utilizado em controles mais avançados não se limitando apenas ao controle com lógica binária.

Com o passar dos anos os CLPs foram se tornando equipamentos de custo reduzido. Para se ter uma ideia, em meados de 1980 o preço do CLP (*hardware*) correspondia a 97% do preço do sistema, próximo a 1990 esse valor passou a 30% e hoje existem casos em que apenas o *software* é dotado de um valor comercial, enquanto que o *hardware* é doado pelo fornecedor. Diz-se então, que o CLP é hoje mais uma ferramenta que pode e deve ser utilizada nos laboratórios de ensino, seja ele em nível técnico ou em nível superior.

3.1 AMBIENTE LABORATORIAL

O Departamento de Engenharia Elétrica (DEE) da FEIS, através do projeto intitulado "Indústria Piloto", participou do programa REENGE, tratando do ensino de automação de processos no curso de Engenharia Elétrica da FEIS.

Com o recurso financeiro obtido pelo programa REENGE, que foi limitado em aproximadamente R\$12.000,00, iniciou-se a construção de seis bancadas, cada uma das quais contendo painéis de circuito de comando e força com os componentes eletromecânicos. A complementação deste laboratório com Controladores Lógicos Programáveis e máquinas elétricas rotativas foi obtida através de obtenção de recursos de outro programa voltado para o



CAPÍTULO 3

IMPLANTAÇÃO DO LABORATÓRIO DE ENSINO DE AUTOMAÇÃO INDUSTRIAL

Como descrito no capítulo anterior, nota-se que a utilização do CLP em vários setores é uma maneira eficiente para automatizar um processo. Embora ele possa ser utilizado em várias aplicações, o setor onde ele causou maior impacto e alcançou maior sucesso, foi na indústria manufatureira, que o adotou como a melhor ferramenta de controle de processo. Por ser capaz de trabalhar com variáveis analógicas o CLP começou a ser utilizado em controles mais avançados, não se limitando apenas ao controle com lógica binária.

Com o passar dos anos os CLPs foram se tornando equipamentos de custo reduzido. Para se ter uma idéia, em meados de 1980 o preço do CLP (*hardware*) correspondia a 97% do preço do sistema, próximo a 1990 esse valor passou a 50% e hoje existem casos em que apenas o *software* é dotado de um valor comercial, enquanto que o *hardware* é doado pelo fornecedor. Diz-se então, que o CLP é hoje mais uma ferramenta que pode e deve ser utilizada nos laboratórios de ensino, seja ele em nível técnico ou em nível superior.

3.1 AMBIENTE LABORATORIAL

O Departamento de Engenharia Elétrica (DEE) da FEIS, através do projeto intitulado “Indústria Piloto”, participou do programa REENGE, tratando do ensino de automação de processos no curso de Engenharia Elétrica da FEIS.

Com o recurso financeiro obtido pelo programa REENGE, que foi limitado em aproximadamente R\$12.000,00, iniciou-se a construção de seis bancadas, cada uma das quais contendo painéis de circuito de comando e força com os componentes eletromecânicos. A complementação deste laboratório com Controladores Lógicos Programáveis e máquinas elétricas rotativas foi obtida através de obtenção de recursos de outro programa voltado para o



aperfeiçoamento do ensino de graduação, o PROIN (com apoio da CAPES), num valor total de R\$18.000,00.

Hoje dispõe-se, em funcionamento no laboratório, 6 bancadas, cada uma com 1 painel elétrico, 1 CLP, 1 inversor de frequência, dispositivos eletromecânicos (relés, temporizadores com retardo na energização e desenergização, contatores), ferramentas e 4 motores de indução, além de vários painéis sinópticos para serem usados na simulação dos processos automatizados.

Este laboratório deverá servir não só para a disciplina optativa criada em função do projeto, *"Prática em Controle e Acionamento de Máquinas Elétricas em Processos Industriais"*, como também para outras disciplinas do curso de graduação em engenharia elétrica da FEIS/UNESP, daí o caráter multidisciplinar no uso do laboratório e equipamentos.

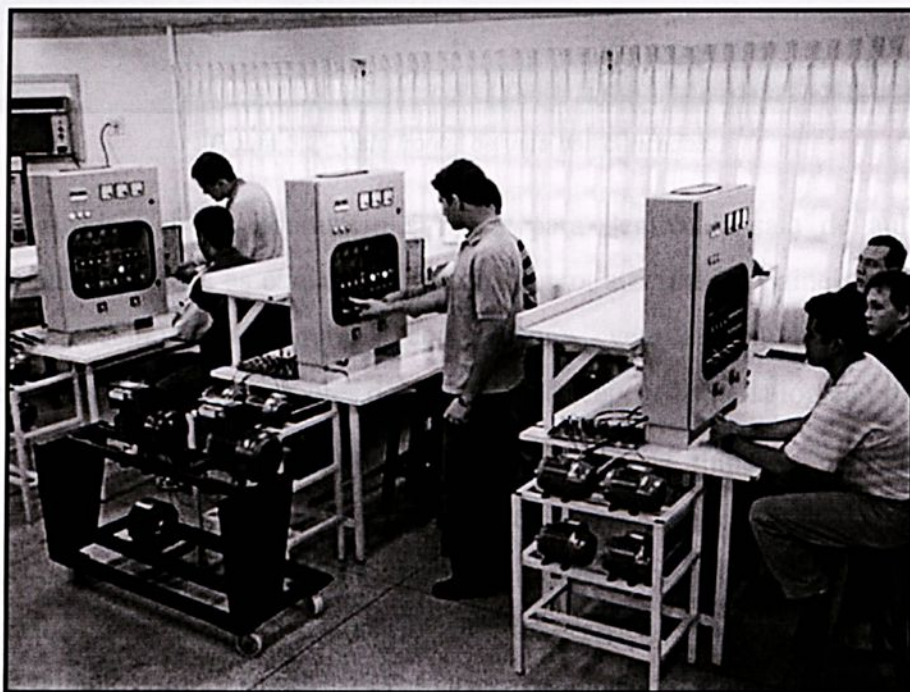


Figura 3.1 - vista geral do laboratório de Automação de Processos: alunos em aula na disciplina *"Prática em Controle e Acionamentos de Máquinas Elétricas em Processos Industriais"*.

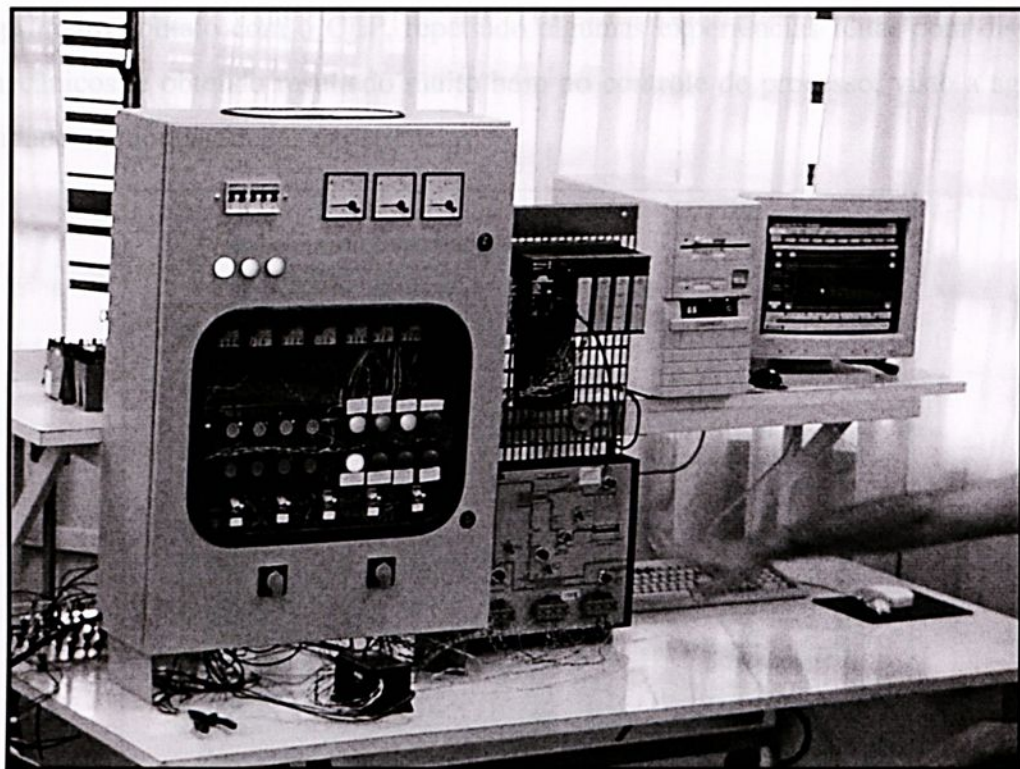


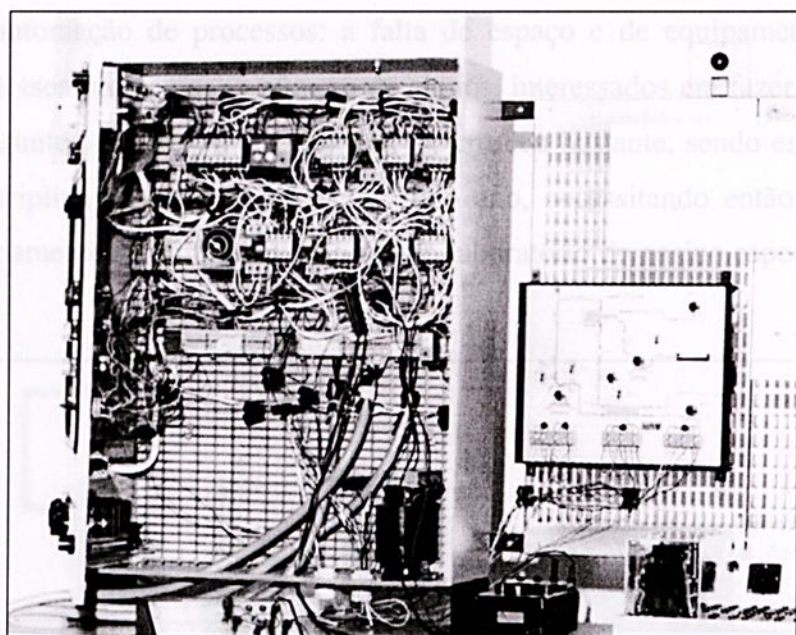
Figura 3.2 - vista frontal de uma das bancadas.

A idéia deste laboratório é trazer para dentro da sala de aula, processos industriais, simulando todas as etapas de funcionamento, através dos painéis sinóticos e dos motores que representam as máquinas (compressores, ventiladores, bombas...) ligando e desligando em configurações diferentes de alimentação, dependendo da lógica programada.

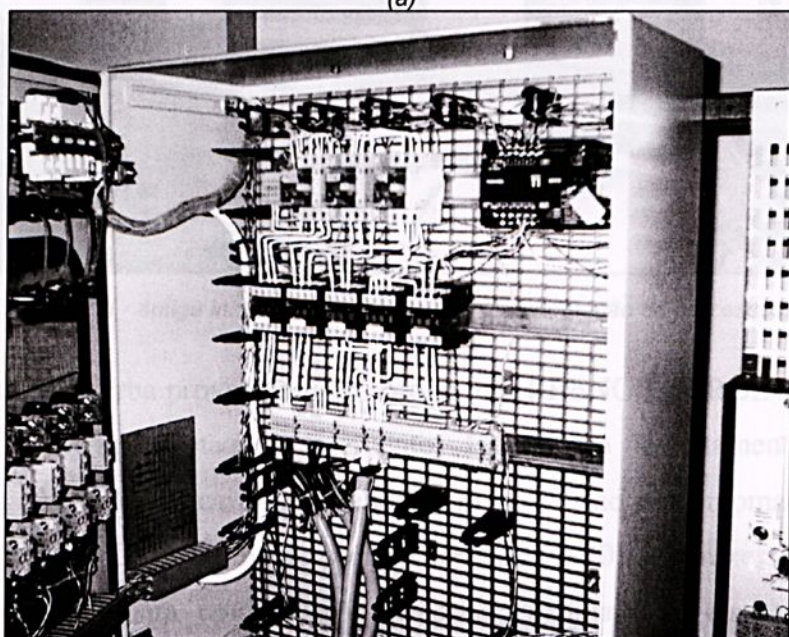
Quando a disciplina foi oferecida a primeira vez, tinha-se apenas uma bancada (painel elétrico, dispositivos eletromecânicos e motores), comprada de Israel, para realização da automação dos processos. Logo 20 alunos e o professor responsável enfrentavam grande problema de falta de espaço, pois o laboratório dividia espaço com o laboratório de máquinas elétricas, e de falta de equipamentos para a realização da mesma.

Em paralelo a essa disciplina alunos de iniciação científica, da FEIS, pesquisavam a viabilidade da utilização de um *micro* CLP para ser usado no ensino, substituindo então os dispositivos eletromecânicos. Após toda formação disciplinar, adquirida em controle de processos, utilizando dispositivos eletromecânicos, com implementação de alguns processos, o *micro* CLP já estava pronto para ser usado na sala de aula. Logo, a primeira turma da disciplina Prática em Controle e Acionamentos de Máquinas Elétricas, já

teve o primeiro contato com o CLP, repetindo algumas experiências feitas com dispositivos eletromecânicos, e obtendo resultado muito bom no controle de processo, visto a agilidade e flexibilidade na montagem das experiências.



(a)



(b)

Figura 3.3 - (a) processo de refrigeração utilizando dispositivos eletromecânicos, (b) processo de refrigeração utilizando CLP.

Essa experiência deu tão certo que o CLP, embora com poucas entradas e saídas, passou a fazer parte do conteúdo programático da disciplina, dividindo-a em 2 etapas:

i) controle de processos utilizando dispositivos eletromecânicos, ii) controle de processos utilizando o CLP.

Ainda assim havia alguns problemas a serem resolvidos para a implantação do laboratório de automação de processos: a falta de espaço e de equipamentos apropriados. Mesmo diante desses problemas, o número de alunos, interessados em fazer essa disciplina e em pesquisar assuntos relacionados à automação, cresceu bastante, sendo esse um indicativo de que essa disciplina deveria ser oferecida todo ano, necessitando então a ampliação no número de equipamentos e da implantação de um laboratório de ensino específico para a área de automação.

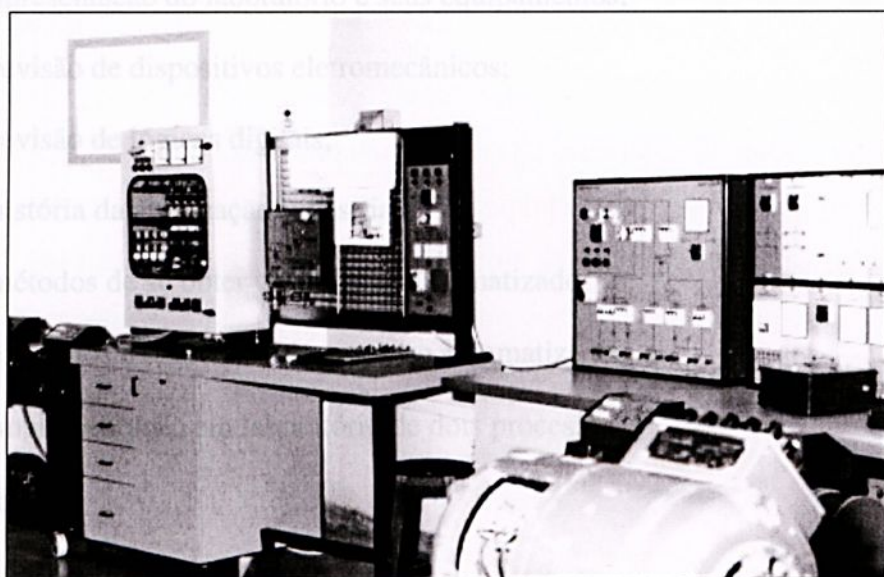


Figura 3.4 - antiga instalação do laboratório de automação de processos.

Logo, a verba proveniente dos programas REENGE e PROIN veio ao encontro às necessidades enfrentadas naquele momento. Fez-se um levantamento do que seria necessário para construção física de um laboratório de ensino em automação, tendo como resultado a aquisição de CLPs de médio porte (*GeFanuc 90-30*), *softwares* de programação e supervisão, dispositivos para comunicação *Ethernet*, manuais, inversores de frequência, gabinetes para montagem do painel elétrico, motores de indução e novos dispositivos eletromecânicos.

O espaço físico foi adquirido no próprio laboratório de ensino de engenharia elétrica, ocupando assim uma das salas do laboratório. Os *layouts* dos painéis elétricos e

sinópticos, dos motores, foram de responsabilidades dos técnicos do Laboratório de Engenharia Elétrica, que utilizaram a oficina mecânica para a construção de tais.

Com a experiência adquirida durante os trabalhos de pesquisa e implantação do laboratório de ensino em automação de processos, é proposto um cronograma de atividades a serem seguidas durante o ensino da disciplina Prática em Controle e Acionamento de Máquinas Elétricas em Processos Industriais. Este cronograma é composto de duas etapas conforme descrito a seguir:

- Primeira Etapa:

- apresentação do laboratório e seus equipamentos;
- revisão de dispositivos eletromecânicos;
- revisão de lógicas digitais;
- história da automação industrial;
- métodos de se obter um sistema automatizado;
- exemplos e exercícios de processo automatizados, e
- implementação em laboratório de dois processos.

- Segunda Etapa:

- apresentação do CLP;
- emprego do CLP e Inversores de Frequência em automação industrial;
- apresentação do *hardware* e suas interfaces;
- comparação entre diferentes modelos e diferentes fabricantes;
- linguagem de programação (generalizando conforme a norma);
- exemplos e exercícios de processos automatizados;
- *software* de programação dos fabricantes de CLP presentes no laboratório;
- implementação em laboratório de dois processos;
- noções de redes de comunicação;



- *software* de supervisão e controle, e
- comunicação de processos em rede com supervisão.

3.2 UTILIZAÇÃO DO LABORATÓRIO NO ENSINO EM GRADUAÇÃO

Para melhor aproveitamento do laboratório na disciplina Prática em Controle e Acionamentos de Máquinas Elétricas em Processos Industriais, são propostas, neste item, através da experiência adquirida pelas equipes de pesquisa, soluções adequadas ao espaço e equipamentos disponíveis no laboratório, de forma a agilizar o desenvolvimento didático da disciplina e oferecer ao aluno uma completa formação no que tange à automação industrial, utilizando dispositivos eletromecânicos e o CLP.

Os pré-requisitos para o aluno cursar tal disciplina, sem comprometer seu desempenho e sua capacidade de assimilar o conteúdo proposto, são dois: Circuitos Digitais e Instalações Elétricas e Industriais. Essas duas disciplinas são importantes, pois o CLP utiliza lógica binária para realizar suas tarefas de controle e, principalmente na primeira etapa do ensino, são necessários conhecimentos de dispositivos eletromecânicos, bem como de configurações diferentes nas ligações dos motores elétricos. Outra importante ferramenta que o aluno deverá estar bem familiarizado é quanto à utilização da informática.

O número ideal de alunos para essa disciplina deverá ser tal que, resulte em três alunos por bancada. Para ministrar o *software* de programação e supervisão, no laboratório de informática (PROIN), o número ideal é de um aluno por micro computador, evitando que alunos se dispersem do objetivo proposto.

3.2.1 Primeira Etapa do Ensino

Nesta etapa, os alunos implementam o controle dos processos, utilizando equipamentos elétricos, nos chamados circuitos de comando eletromecânicos.

É importante frisar que a tecnologia utilizando dispositivos eletromecânicos ainda é utilizada em processos pequenos, onde não são necessárias uma precisão muito grande do controle, nem mesmo uma interconexão de informações, via barramento de dados, dos processos controlados. Também existem, ainda no Brasil, muitas indústrias de médio e até grande porte, em que parte da automação ainda utiliza dispositivos eletromecânicos, sendo



necessário ao engenheiro responsável conhecer tais dispositivos para efetuar manutenção e eventual troca desses por equipamentos de tecnologia mais avançada.

Ainda na primeira etapa é oportuno dizer que dispositivos eletromecânicos estão mais sujeitos a defeitos devido aos contatos mecânicos, por isso, o comissionamento nos painéis elétricos é freqüentemente necessário, acrescentando despesas financeiras e desperdício de tempo.

Outro fato importante de se tratar é quanto ao tempo de chaveamento dos dispositivos eletromecânicos. Esse ponto passaria despercebido sem utilização do CLP, mas como o chaveamento do CLP, e conseqüentemente sua lógica de controle, depende apenas de contatos lógicos, ele é muito mais rápido do que o chaveamento feito por dispositivos eletromecânicos. Assim, alguns intertravamentos utilizando diagrama elétrico (lógica relé) pode não funcionar quando traduzido para lógica de diagrama de contatos (diagrama *Ladder*) utilizada no CLP, podendo acarretar assim curto circuito fase-fase.

Esse problema é verificado na prática e pode ser corrigido atrasando, na lógica de programação do CLP, o comando de alimentação de dispositivos (geralmente contadores), através do módulo de saída do CLP, que poderiam provocar o curto-circuito, ou deixando de implementar esse intertravamento através da lógica, usando os contatos auxiliares dos contadores para executar tal intertravamento; visto que isso pode ser feito de maneira muito fácil, não resultando num painel elétrico cheio de fios, nem tomando muito tempo na sua execução.

As aulas teóricas podem ser ministradas através de transparências. O aluno recebe no final de cada aula o material que será ministrado na próxima, fazendo com que o ensino fique mais dinâmico.

Após toda aula prática é importante a apresentação de um relatório simples, relatando os objetivos, dificuldades e soluções encontradas nas experiências. Nas aulas práticas é sempre importante a presença de um técnico, que auxiliará o professor durante a execução do projeto. O técnico acompanhará de perto, principalmente durante as etapas iniciais de montagem. Aos poucos o técnico e o professor responsável deixam os alunos mais à vontade para motivar neles o espírito criativo e aumentar a confiança na realização de sua experiência.



Após implementação do controle, através dos dispositivos eletromecânicos, nenhum processo deverá ser ligado sem o parecer do professor ou técnico responsável. Para verificar se toda lógica de controle está funcionando corretamente, deve-se desligar o circuito de força, que aciona os motores, os quais simulam a operação do processo. Tomadas essas providências já pode-se ligar o processo e verificar se os contatores atracam conforme desejado na lógica de controle. Verificado o funcionamento correto da lógica, o professor ou técnico poderá autorizar os alunos a ligarem a força, através dos relés de sobrecarga, e todo o processo para verificar se os motores estão funcionando corretamente, simulando assim um processo real. Se mesmo tomadas essas providências ocorrer um curto entre fases, os alunos da bancada devem verificar o que ocasionou o problema, descrevendo-o no relatório.

Quanto à execução das experiências é importante que o professor acompanhe de perto cada equipe para que todos tenham a iniciativa de participar diretamente da montagem do processo, não ficando assim nenhum aluno disperso do objetivo proposto na aula. Uma forma de incentivar os alunos a interagirem ao processo, é uma boa preparação pré-aula do professor, do tipo perguntas pré-formuladas para fazer a um aluno, aleatoriamente, durante a execução da experiência. Por exemplo: *“qual a lâmpada que sinaliza o compressor do ar condicionado ligado?”*.

Após a montagem do processo a ser automatizado por dispositivos eletromecânicos, o professor aplica uma prova sobre o conteúdo ministrado durante a primeira etapa do ensino. Para finalizar esta etapa o professor aplica um projeto aberto e extra-classe, onde o aluno deve pesquisar uma planta a ser automatizada, descrevendo os objetivos, soluções e dificuldades deste projeto. Isso é muito interessante, pois indica o nível de aprendizado do aluno, bem como pode vir a ser contribuição para a disciplina, como material didático a ser utilizado na próxima turma do curso.

3.2.2 Segunda Etapa do Ensino

Nesta etapa, quando os alunos já estão conhecendo a metodologia de controle utilizando dispositivos eletromecânicos, o professor terá maior facilidade de mostrar como funciona e onde é empregado o CLP.



As aulas teóricas podem ser apresentadas utilizando transparências, assim como na primeira etapa, também dinamizando as aulas e otimizando o tempo de exposição.

Um seminário bastante simples, com exposição programada de 15 minutos, por equipe, é uma boa opção para dinamizar o início da segunda etapa de ensino. Isso pode ser dado em uma aula e cada equipe estará responsável pela pesquisa e apresentação de um CLP de um determinado fabricante, expondo de maneira resumida suas características e funcionamento. O resultado deste seminário, que pode ser chamado de mini fórum de discussão, serve como base para as aulas seguintes.

Ao ensinar o *software* de programação é fundamental cada aluno ter um microcomputador e contar com o auxílio de monitores, que podem ser alunos de iniciação científica, estagiário ou alguém que desenvolve trabalho de formatura na área. Outra ferramenta importante e fundamental para ministrar o *software* do CLP é a utilização de um projetor multimídia. Assim todos alunos poderão acompanhar em tempo-real o conteúdo exposto pelo professor.

No laboratório a primeira atividade do aluno será testar todos os módulos de entrada e saída do CLP, com testes bastante simples, conforme será exposto no item 4.1. Após o teste dos módulos o professor expõe os processos a serem automatizados; é interessante cada equipe montar dois processos diferentes para se familiarizar bastante com o *hardware* e *software*.

Os mesmos cuidados, àqueles tomados na primeira etapa devem ser observados, ou seja, um técnico deve acompanhar e auxiliar o professor durante as aulas e os alunos não poderão ligar o processo sem que antes haja um parecer do professor ou técnico responsável.

Terminado todas as conexões elétricas dos pontos de entrada e saída (E/S) do CLP, para verificar se a lógica transferida ao CLP está funcionando corretamente, desliga-se os relés de sobrecarga das máquinas do processo e liga-se o processo automatizado para verificar se os contadores estão atracando corretamente e se as lâmpadas estão ligadas conforme a lógica programada. Estando tudo correto, pode-se ligar os relés e verificar se o processo está funcionando corretamente. Após cada experiência deve-se cobrar um relatório sucinto e objetivo do processo controlado através do CLP.



Terminadas as montagens dos processos o professor aplica uma prova que avaliará os conhecimentos do aluno a respeito do conteúdo ministrado e a sua capacidade criativa para solucionar problemas de automação de processos. Também pode-se propor que os alunos transformem a lógica de controle do trabalho extra-classe, da primeira etapa de ensino, para a linguagem *Ladder*.

3.2.3 Avaliação e Conceito Final

A avaliação desta disciplina deve conter perguntas com formulação teórica, mas principalmente questões que exigem do aluno raciocínio lógico e conhecimento técnico. Propõe-se que sejam aplicadas duas avaliações, quatro relatórios de experiências em laboratório, um seminário, dois trabalhos extra-classe, mais uma nota de participação em aula. A média final do aluno será obtida conforme a equação a seguir:

$$MF = \left(\frac{A_1 + A_2}{2} \right) * 0,5 + \left(\frac{R_1 + R_2 + R_3 + R_4}{4} \right) * 0,2 + \left(\frac{E_1 + E_2}{2} \right) * 0,2 + \left(\frac{S + P}{2} \right) * 0,1 \quad (1)$$

onde,

MF = média final;

A = avaliação;

R = relatório;

E = trabalho extra-classe;

S = seminário, e

P = participação em aula.

3.3 PROPOSTA DE LABORATÓRIO DE PESQUISA EM AUTOMAÇÃO

Neste item sugere-se a construção de um novo laboratório de pesquisa em automação, que pode vir a ocorrer através de um projeto amplo com algum órgão de fomento ou através de parceria com alguma empresa nacional ou multinacional interessada na pesquisa em automação industrial ou residencial.



3.3.1 Justificativa

Para que o ensino esteja sempre em sintonia com os últimos acontecimentos tecnológicos no ramo de automação industrial, é necessário que se mantenha, em constante desenvolvimento, um grupo de pesquisa para atualizar o ensino toda vez que novas tendências e tecnologias forem surgindo no mercado. A disciplina intitulada “Prática em Controle e Acionamento de Máquinas Elétricas em Processos Industriais”, tem funcionado desta forma, porém os pesquisadores estão dividindo espaço e equipamentos, num mesmo laboratório, com os alunos desta disciplina, atrapalhando ambas atividades, pois nem o ensino e nem a pesquisa podem utilizar todo o material disponível no laboratório.

Neste último semestre pesquisadores, em nível de iniciação científica, estágio, trabalho de formatura e mestrado, utilizaram parte do laboratório para desenvolver pesquisas, sobre a utilização de CLPs em rede e supervisão, que ainda não foram incorporadas no conteúdo da disciplina. Com isso tanto a disciplina quanto a pesquisa só puderam utilizar a metade dos equipamentos disponíveis no laboratório, prejudicando mais a disciplina do que a pesquisa. Finalizado o estudo em redes e supervisão de CLPs, os pesquisadores deixarão o laboratório para uso somente da disciplina, visto que esta é prioridade em relação à formação dos alunos, fazendo com que a pesquisa em automação de processos perca seu espaço.

Portanto, chega-se a conclusão que é essencial um laboratório destinado somente à pesquisa, que deverá ser implantado para que pesquisadores em automação industrial possam desenvolver suas pesquisas direcionadas para incorporação no ensino, mantendo sempre atualizado o ensino e investigando possibilidades de desenvolver novas tecnologias que poderão ser incorporados no mercado e no ramo de automação industrial e residencial.

3.3.2 O Laboratório

Segue então uma sugestão de equipamentos e espaço necessário para a construção deste laboratório:

- Espaço físico de 100 m²;
- 5 bancadas com painéis elétricos;



- 5 computadores PC com placa de rede;
- 5 CLPs modulares com cartão de simulador, *Ethernet*, E/S digitais e analógicas, com cabos de comunicação à rede e ao microcomputador;
- um *Hub* para testes em novas configurações de rede;
- dispositivos *FLEX I/O*, para controle de E/S distribuídas;
- sensores inteligentes (termostatos, pressostatos, ultra-som...), que se conectam em um nível de rede de Chão de Fábrica (*DeviceNET*);
- sondas para utilização em automação de processos químicos, como tratamento de água;
- 1 IHM (Interface Homem-Máquina);
- *software* de programação e supervisão;
- licença para utilização da *WebView*, habilitando o controle via internet, e
- motores de indução para simulação de processos ou implementação de protótipos.

Com esses equipamentos pode-se utilizar todos recursos do CLP, bem como estudar novas ferramentas em sua utilização. Também pode-se estudar o controle numa rede de CLPs conectadas à *Web*, verificando os problemas que poderiam acontecer neste tipo de comunicação, como por exemplo, atraso no controle (*Delay time*).

Com os equipamentos disponíveis no laboratório de ensino pode-se executar muitas tarefas de controle e, para iniciar a utilização do CLP, é mostrado no próximo capítulo algumas sugestões de experiência para o aluno utilizar durante a segunda etapa de ensino (item 3.2.2).

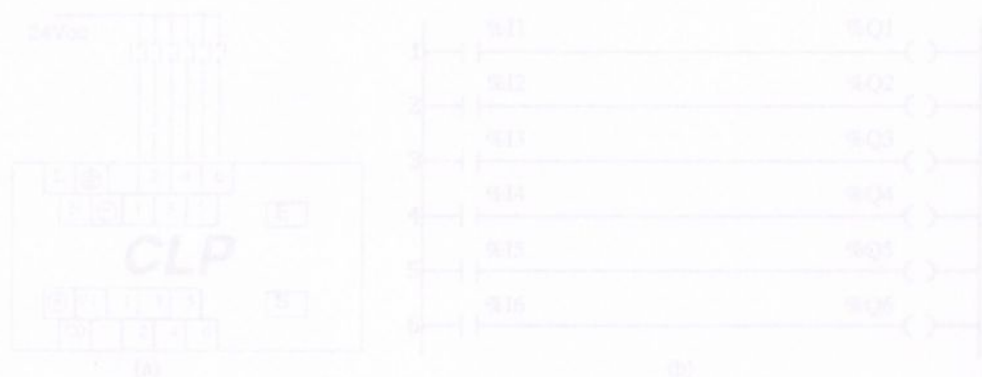


Figura 4.1 - (a) Visão física das chaves (b) Tabela de programação

SUGESTÕES DE EXPERIÊNCIAS

Sugere-se, neste item, algumas experiências novas para o ensino de automação industrial, utilizando o CLP como principal solução dos objetivos de controle, sendo que estas podem ser utilizadas como motivação para aprendizagem, pois são experiências simples, mostrando parte da capacidade de um CLP.

Após familiarização com programação em diagrama *Ladder* (ver Apêndice A), o aluno estará apto a implementar algumas experiências básicas com o CLP. A seguir são propostas algumas dessas experiências.

4.1 EXPERIÊNCIA 1: TESTE DAS E/S DO CLP

A primeira experiência do aluno no laboratório de automação frente ao CLP é testar se todas as entradas e saídas do equipamento estão funcionando corretamente. Supondo um CLP com 6 entradas de 24Vcc e 6 saídas de 127Vac, podemos proceder da seguinte forma para testar as E/S.

Primeiro cria-se um barrento de 24 Vcc, onde será ligado o comum das 6 chaves. Escreva um programa *Ladder*, conforme a figura abaixo, e ligue uma chave por vez verificando se as saídas correspondentes estão sendo alimentadas com 127Vac.

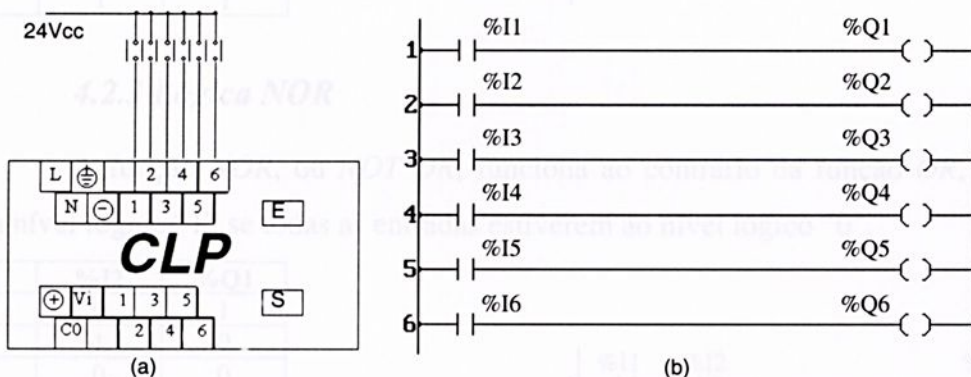


Figura 4.1 - (a) ligação física das chaves (b) lógica de programação.

Confirmado que todas entradas e saídas do CLP estão funcionando corretamente, pode-se utilizar o equipamento para atuar em algum processo.

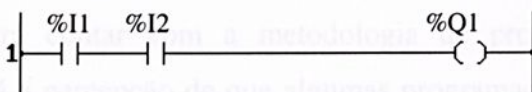
4.2 EXPERIÊNCIA 2: LÓGICA AND, OR, NAND, NOR

Esta experiência trata do princípio básico de programação do CLP: as funções lógicas, onde utiliza-se mais de uma entrada e apenas uma saída para verificar o estado da saída. Considera-se nível lógico “1” sempre que a entrada ou saída estiver energizada. Para facilitar essa experiência utiliza-se apenas 2 entradas e 1 saída.

4.2.1 Lógica AND

Sabe-se que na lógica *AND* se uma das entradas for igual a zero a saída também será zero. As entradas são %I1 e %I2, e a saída é %Q1. Segue a tabela verdade para essa lógica e a programação em Diagrama *Ladder* para a solução desta lógica.

%I1	%I2	%Q1
0	0	0
0	1	0
1	0	0
1	1	1



4.2.2 Lógica OR

Na lógica *OR*, estando uma das entradas do CLP em nível lógico “1”, a saída deverá ser “1”.

%I1	%I2	%Q1
0	0	0
0	1	1
1	0	1
1	1	1



4.2.3 Lógica NOR

A função *NOR*, ou *NOT OR*, funciona ao contrário da função *OR*, ou seja, a saída tem nível lógico “1” se todas as entradas estiverem ao nível lógico “0”.

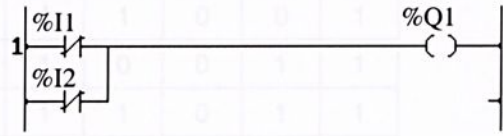
%I1	%I2	%Q1
0	0	1
0	1	0
1	0	0
1	1	0



4.2.4 Lógica NAND

A função *NAND*, ou *NOT AND*, funciona ao contrário da função *AND*, ou seja, a saída tem nível lógico “1” se pelo menos uma das entradas tiver nível lógico “0”.

%I1	%I2	%Q1
0	0	1
0	1	1
1	0	1
1	1	0



Conhecendo as formas de programação para cada função lógica, podemos obter soluções para casos em que necessita-se combinar mais de uma função, por exemplo, para obter a solução de uma grande tabela da verdade.

4.3 EXPERIÊNCIA 3: APLICAÇÃO DA LÓGICA PARA UM DECODIFICADOR DE SETE SEGMENTOS

Nesta experiência o aluno poderá contar com a metodologia de projeto assimiladas na disciplina Circuitos Digitais e terá a percepção de que algumas programações poderão ser sistematizadas com esta técnica.

Conforme visto na figura a seguir, nesta experiência têm-se 4 entradas e 7 saídas, sendo uma para cada segmento.

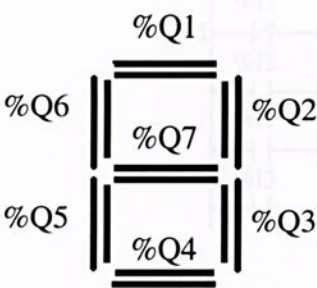


Figura 4.2 – Display de Sete segmentos.

Entrada BCD					Sete Segmentos						
D	%I1	%I2	%I3	%I4	%Q1	%Q2	%Q3	%Q4	%Q5	%Q6	%Q7
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

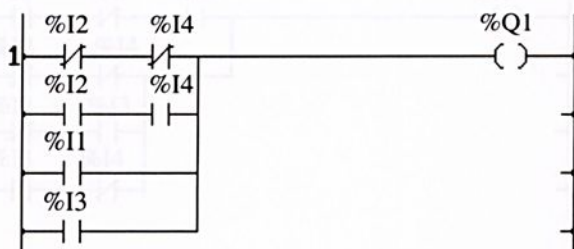
Resolve-se tal experiência tratando das saídas separadamente. Uma forma simples de resolver esse problema é aplicando a minimização por mapas de *Karnaugh*.

A solução deste problema é alcançada em partes. Obtém-se uma lógica para cada saída isoladamente, depois junta-se todas em um único programa, que executará a decodificação da entrada BCD para a saída de sete segmentos. Por simplificação chamaremos as entradas %I1, %I2, %I3 e %I4 de A, B, C e D, respectivamente.

- Saída %Q1

	$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
$\overline{C}\overline{D}$	00	01	11	10
$\overline{C}D$	00	0	X	1
$\overline{C}D$	01	0	1	X
CD	11	1	1	X
$C\overline{D}$	10	1	1	X

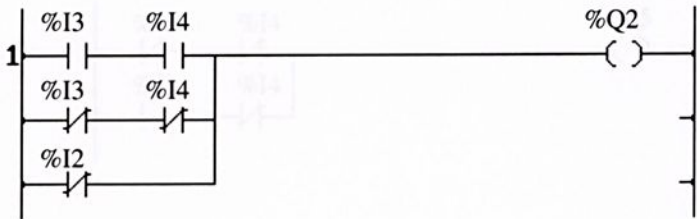
$$\%Q1 = A + C + BD + \overline{B}\overline{D}$$



- Saída %Q2

		$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
		00	01	11	10
$\overline{C}\overline{D}$	00	1	1	X	1
$\overline{C}D$	01	1	0	X	1
CD	11	1	1	X	X
$C\overline{D}$	10	1	0	X	X

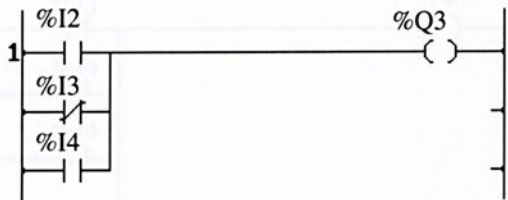
$\%Q2 = CD + \overline{C}\overline{D} + \overline{B}$



- Saída %Q3

		AB	$A\overline{B}$	$\overline{A}\overline{B}$	$\overline{A}B$
		00	01	11	10
CD	00	1	1	X	1
$C\overline{D}$	01	1	1	X	1
$\overline{C}\overline{D}$	11	1	1	X	X
$\overline{C}D$	10	0	1	X	X

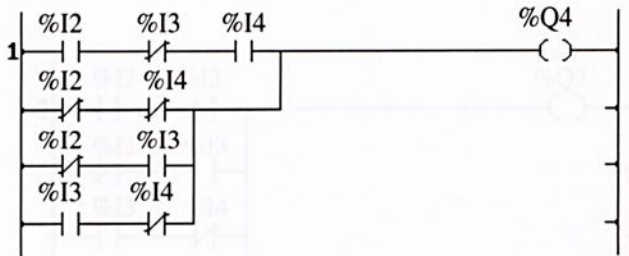
$\%Q3 = B + \overline{C} + D$



- Saída %Q4

		$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
		00	01	11	10
$\overline{C}\overline{D}$	00	1	0	X	1
$\overline{C}D$	01	0	1	X	0
CD	11	1	0	X	X
$C\overline{D}$	10	1	1	X	X

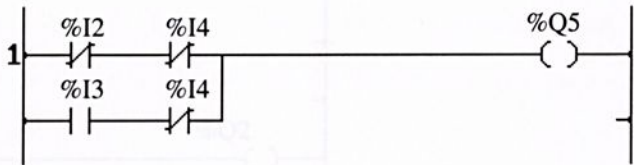
$\%Q4 = \overline{C}\overline{D} + \overline{B}\overline{C}D + \overline{B}C + \overline{B}\overline{D}$



- Saída %Q5

		$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
		00	01	11	10
$\overline{C}\overline{D}$	00	1	0	X	1
$\overline{C}D$	01	0	0	X	0
CD	11	0	0	X	X
$C\overline{D}$	10	1	1	X	X

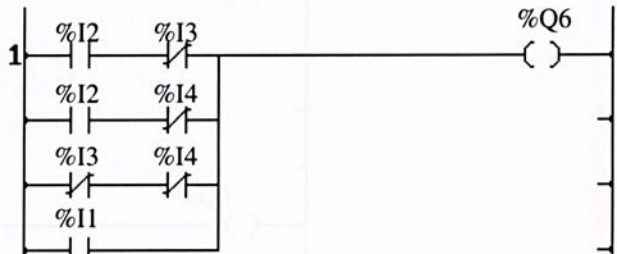
$\%Q5 = C\overline{D} + \overline{B}\overline{D}$



- Saída %Q6

		$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
		00	01	11	10
$\overline{C}\overline{D}$	00	1	1	X	1
$\overline{C}D$	01	0	1	X	1
CD	11	0	0	X	X
$C\overline{D}$	10	0	1	X	X

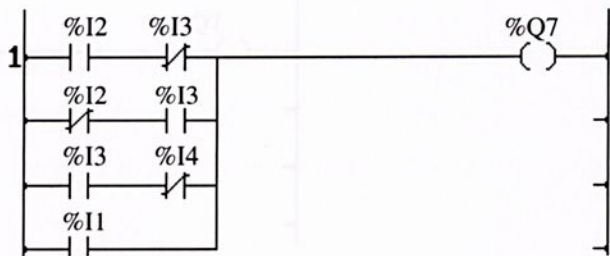
$\%Q6 = A + \overline{C}\overline{D} + B\overline{C} + B\overline{D}$



- Saída %Q7

		$\overline{A}\overline{B}$	$\overline{A}B$	AB	$A\overline{B}$
		00	01	11	10
$\overline{C}\overline{D}$	00	0	1	X	1
$\overline{C}D$	01	0	1	X	1
CD	11	1	0	X	X
$C\overline{D}$	10	1	1	X	X

$\%Q7 = A + C\overline{D} + B\overline{C} + \overline{B}C$



O resultado então, para uma forma de programação *Ladder*, de decodificação do código BCD para sete segmentos, é mostrado a seguir.

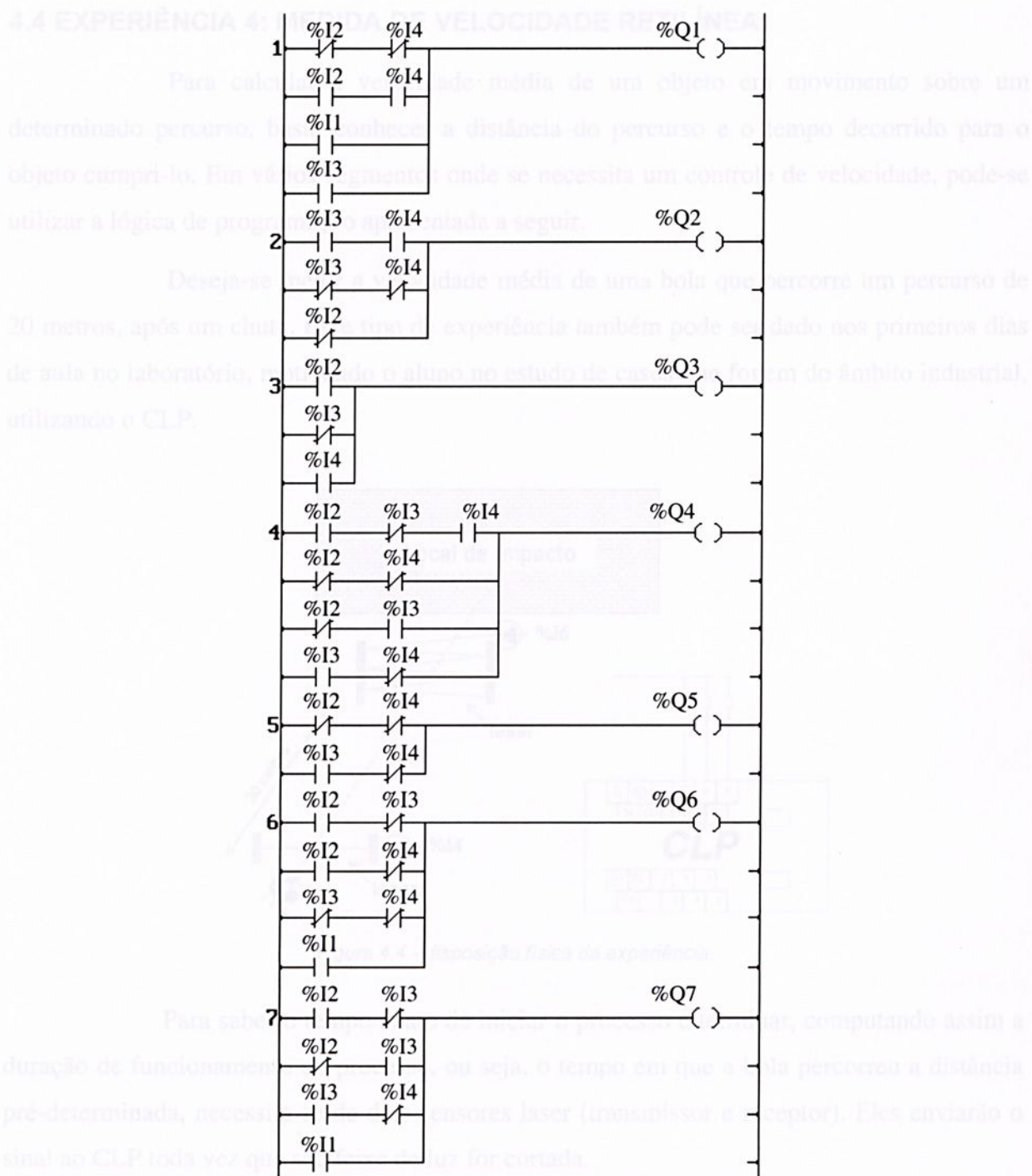


Figura 4.3 - diagrama Ladder para o programa decodificador.

São várias as possibilidades de construção, via programação *Ladder*, de lógicas digitais, utilizando todo o conhecimento adquirido em Circuitos Digitais 1. Várias outras sugestões de experiências podem ser apresentadas, contudo a apresentada, neste item 4.3, engloba vários conhecimentos.

4.4 EXPERIÊNCIA 4: MEDIDA DE VELOCIDADE RETILÍNEA

Para calcular a velocidade média de um objeto em movimento sobre um determinado percurso, basta conhecer a distância do percurso e o tempo decorrido para o objeto cumpri-lo. Em vários segmentos onde se necessita um controle de velocidade, pode-se utilizar a lógica de programação apresentada a seguir.

Deseja-se medir a velocidade média de uma bola que percorre um percurso de 20 metros, após um chute. Esse tipo de experiência também pode ser dado nos primeiros dias de aula no laboratório, motivando o aluno no estudo de casos que fogem do âmbito industrial, utilizando o CLP.

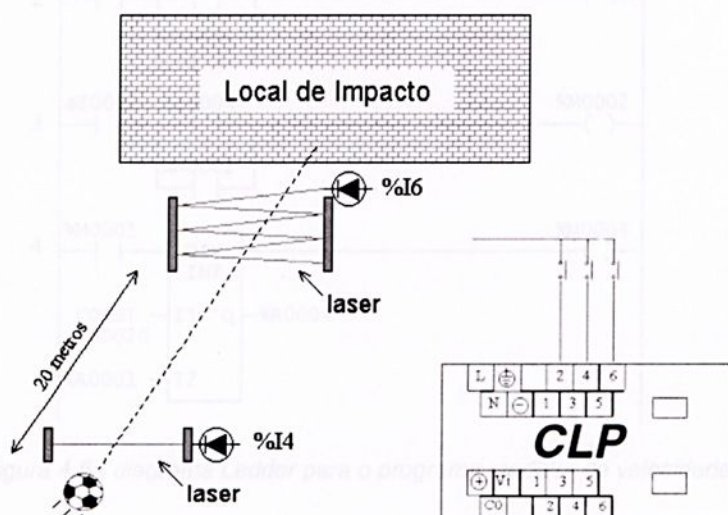


Figura 4.4 - disposição física da experiência.

Para saber o tempo exato de iniciar o processo e terminar, computando assim a duração de funcionamento do processo, ou seja, o tempo em que a bola percorreu a distância pré-determinada, necessita-se de dois sensores laser (transmissor e receptor). Eles enviarão o sinal ao CLP toda vez que seu feixe de luz for cortado.

O primeiro sensor será colocado logo à frente da bola, mandando o sinal responsável para dar início ao processo, e o segundo será colocado a 20 metros de distância para calcular o tempo do percurso da bola. O segundo laser transmite o sinal até o seu receptor, através de um jogo de espelhos para que ele atue em uma região maior, diminuindo o risco da bola passar por um local onde o feixe laser não será cortado. Uma chave geral de liga



e desliga, que também serve como *reset* para todo o sistema, é inserida na entrada %I2 do CLP.

O programa lógico para calcular o tempo e assim a velocidade da bola é dado conforme a figura a seguir.

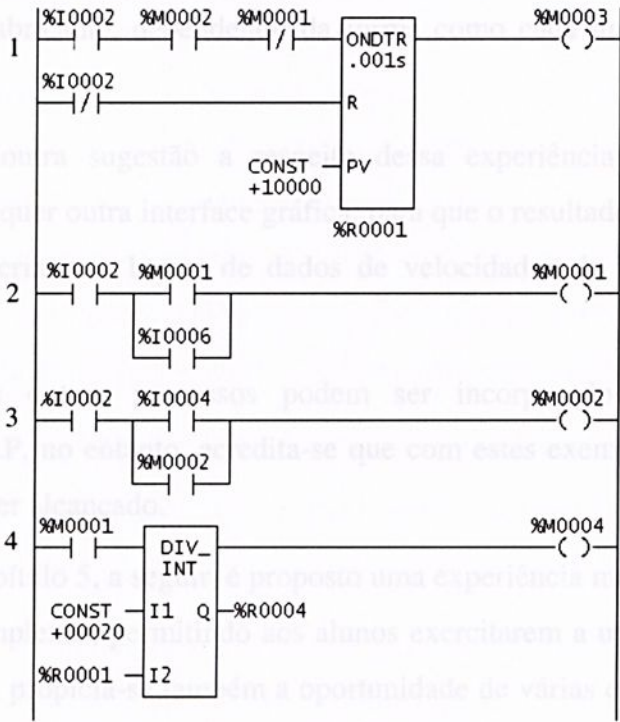


Figura 4.5 - diagrama Ladder para o programa medidor de velocidade.

Ligando a Chave Geral (%I2), retira-se o *reset* do temporizador referente ao registro (%R1), conforme visto na *rung* 1, habilitando a sua função de contagem. Quando a bola passar pelo primeiro sensor, este enviará um pulso na entrada %I4 do CLP que, de acordo com a *rung* 3, energizará o bit de controle interno %M2. Na *rung* 1, observa-se que o temporizador referente ao registro %R1 começa a contar. Quando a bola corta o feixe de luz do segundo sensor, este envia um sinal à entrada %I6 do CLP que, conforme a *rung* 2, energizará o bit de controle interno %M1. Esse bit corta o fluxo de energia do temporizador (*rung* 1), fazendo com que a contagem pare.

Na *rung* 4 é habilitado o bloco de divisão *DIV_INT*, que divide a entrada I1 (20 metros) pela entrada I2 (tempo decorrido do temporizador %R1), guardando o resultado no registro %R4. O bit de controle %M4 indica o fim do processo.



Embora pareça simples, esse tipo de experiência exige bastante destreza dos projetistas, pois eles terão que construir os sensores laser e protegê-los para que uma possível pancada da bola não os danifique. Outra importante observação é que o CLP deve estar em local onde o sinal emitido por tais sensores não se perca ou demore a chegar, pois o evento é muito rápido e não pode haver atraso na comunicação. A lógica a ser programada pode mudar de fabricante para fabricante, dependendo da forma como cada um executa suas operações matemáticas.

Uma outra sugestão a respeito dessa experiência é utilizar um *software* supervisorio, ou qualquer outra interface gráfica, para que o resultado seja mostrado na tela do computador e para criar um banco de dados de velocidades da bola para vários eventos (chutes).

Vários outros processos podem ser incorporados nesta fase inicial de familiarização ao CLP, no entanto, acredita-se que com estes exemplos propostos o objetivo da fase inicial pode ser alcançado.

No capítulo 5, a seguir, é proposto uma experiência mais completa, com lógicas de controle mais complexas, permitindo aos alunos exercitarem a utilização de CLPs ligados em rede. Além disso, propicia-se também a oportunidade de várias equipes interagirem para a realização de um objetivo comum: o funcionamento de um processo, de várias etapas, para produção de um único produto.



Figura 5.1 - proposta final: equipamentos ligados em rede

Primeiramente os processos são implementados individualmente, para depois ligá-los em rede, supervisionados pelo *software* de supervisão e controle. Todos os processos serão descritos a seguir, acompanhados de sua programação *Ladder*.

PROPOSTA FINAL: UMA INDÚSTRIA

Para aplicação dos conceitos, teorias e novas visões tecnológicas adquiridos durante a pesquisa, e com o objetivo de incorporar ao ensino, é simulado em laboratório uma pequena indústria, utilizando os equipamentos disponíveis em laboratório. Essa pequena indústria é dividida em quatro segmentos: 1) Processo de Mistura e Aquecimento de Líquidos, 2) Controle de Nível de Líquido, 3) Processo de Refrigeração e 4) Sinalização através de um sistema de Pêndulo Invertido. Esses quatro segmentos são controlados por 3 CLPs:

- CLP1 – responsável pelo processo de refrigeração e sinalização via pêndulo invertido;
- CLP2 – responsável pelo processo de mistura e aquecimento de líquidos, e
- CLP3 – responsável pelo controle de nível de líquido.

Todos os segmentos estão ligados em rede e supervisionados pelo *software* de supervisão *Cimpliticy-HMI* da *GeFanuc*. Dessa forma, pode-se intertravar todos segmentos da indústria que estão ligados na mesma rede.

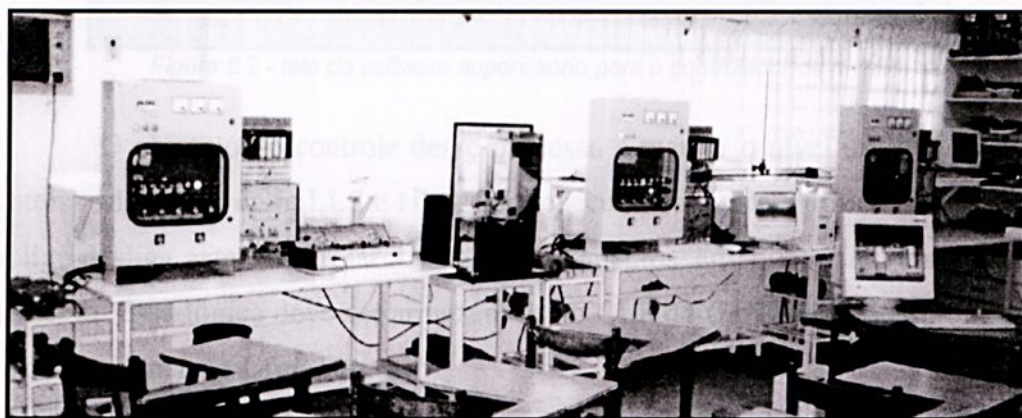


Figura 5.1 - proposta final: equipamentos ligados em rede.

Primeiramente os processos são implementados individualmente, para depois ligá-los em rede, supervisionados pelo *software* de supervisão e controle. Todos os processos serão descritos a seguir, acompanhados de sua programação *Ladder*.

No curso proposto, cada equipe poderá desenvolver e implementar um desses processos e, no final, com interação entre as mesmas, poderão conectar em rede e executar o monitoramento e controle via micro computador.

Ao final desta etapa os alunos terão assimilado praticamente todos os recursos do CLP. Mais detalhes, sobre os processos da proposta final do curso, encontra-se nos apêndices B e C, sendo que neste capítulo 5 não serão apresentadas, por conveniência, as etapas de aquecimento de líquido do CLP2 e de sinalização via pêndulo invertido do CLP1, facilitando o entendimento dos diagramas *Ladder*.

5.1 CONTROLADOR DE NÍVEL (CLP3)

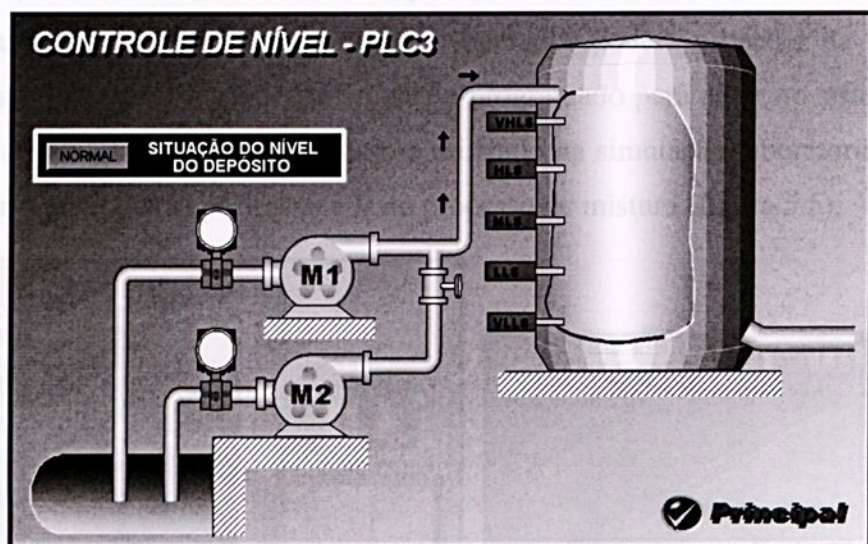


Figura 5.2 - tela do software supervisorio para o controlador de nível.

O objetivo do controle desse processo é manter o nível do líquido, bombeado pelos motores M1 e M2, entre LLS e HLS. Para executar essa tarefa utiliza-se a lógica binária do tipo liga-desliga para controlar os exatos momentos em que os motores entram em funcionamento. Essa lógica deve ser armazenada no CLP da *GeFanuc* série 90-30, através do *software* de programação *LogicMaster*.

De forma bastante geral, esse sistema funciona da seguinte forma: quando o nível do líquido estiver em operação normal, ou seja, entre LLS e HLS, apenas um motor estará funcionando; se por acaso, o nível do líquido, no depósito, estiver abaixo de LLS o segundo motor entra em funcionamento, fazendo com que o nível do líquido volte a subir. Ao

atingir o nível alto HLS os motores param de funcionar. Para evitar que um motor funcione mais tempo que outro, nas operações normais, causando um desgaste maior sobre ele, na lógica de programação é colocado um comando que alterna o funcionamento entre eles.

Os sensores VLLS e VHLS são usados como alarme e segurança. Quando o nível do líquido atinge o sensor VLLS ele indica uma falha e desliga os motores. Da mesma forma, quando o nível do líquido atingir VHLS ele também indica uma falha e pára o sistema, sendo que este só volta a funcionar após execução de um comando do tipo *reset*.

O comando *reset* pode ser dado de duas formas: localmente ou remotamente. Estando o operador ao lado do processo, ao ver a lâmpada sinalizando falha, este localizará e solucionará o problema e, após isto, poderá pressionar uma botoeira para executar o comando *reset*, rearmando o processo. Remotamente o operador poderá proceder da mesma forma, porém utilizando o *software* supervisor, que é programado para atuar no sistema mandando o sinal *reset* através da rede. Este processo é utilizado na simulação laboratorial da indústria, para controlar o nível dos depósitos A e B do processo de mistura (figura 5.5).

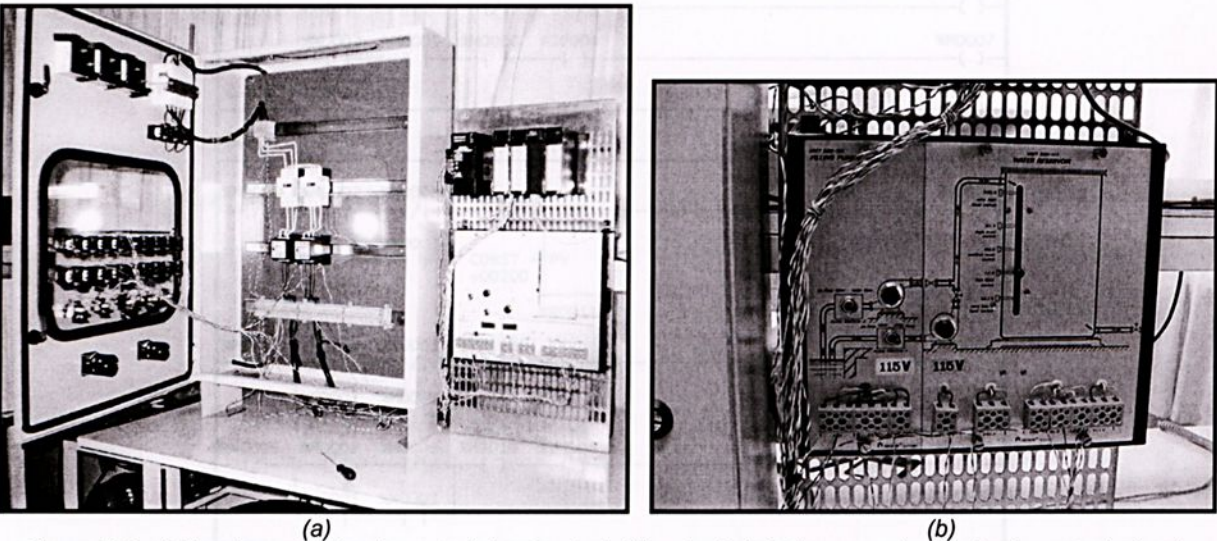


Figura 5.3 - (a) implementação do controlador de nível, (b) painel sinóptico para simulação do controlador de nível.

A figura a seguir descreve o programa lógico, em linguagem *Ladder*, para o controlador de nível. No apêndice B é apresentada, em detalhes, esta mesma lógica em diagrama elétrico e *Ladder*, descrevendo os passos de seu funcionamento e inserindo novos comandos para comunicação e intertravamento com outros processos em rede, por isso para melhor entendimento sugere-se ler a posterior o apêndice B desta monografia.



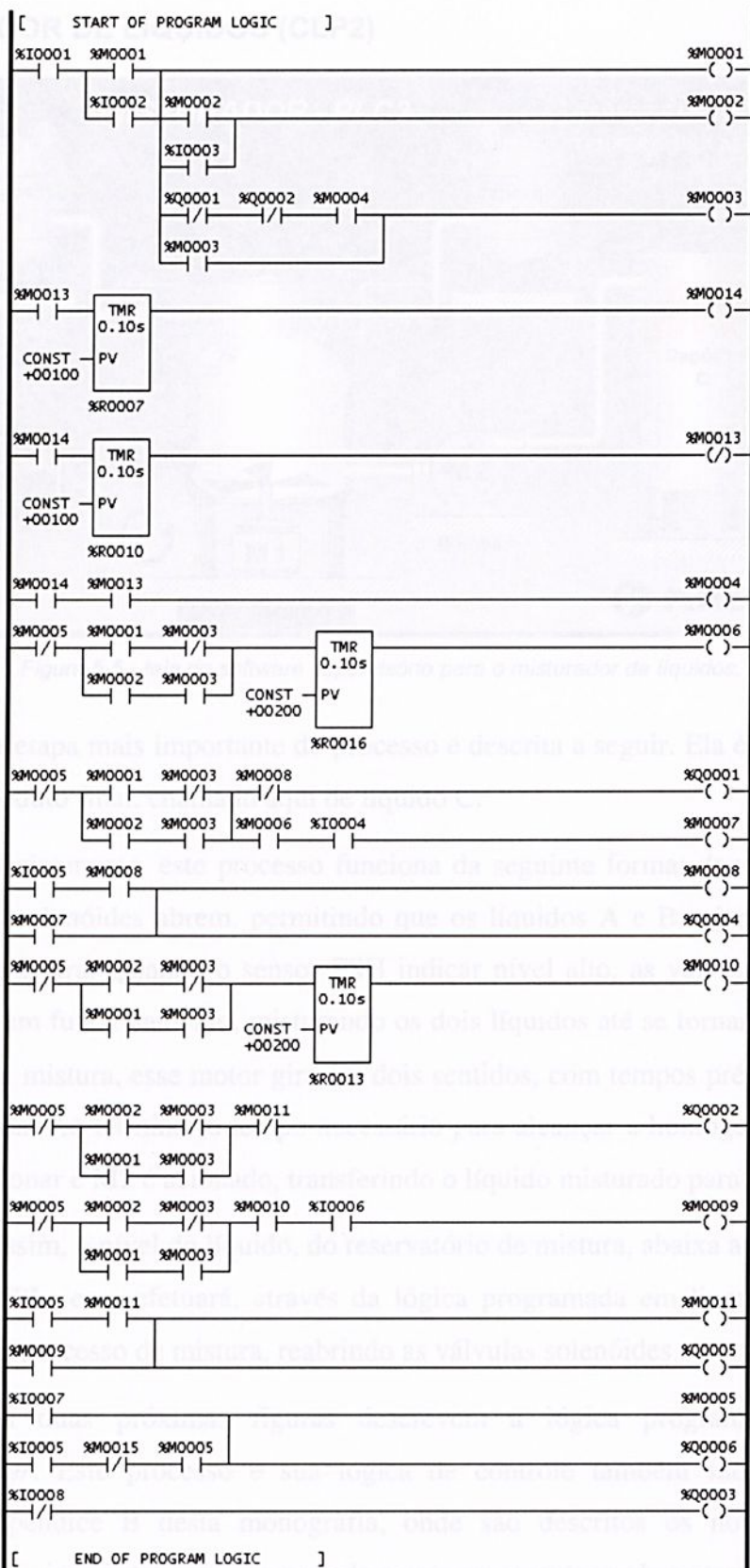


Figura 5.4 - lógica de programação Ladder para o controlador de nível.

5.2 MISTURADOR DE LÍQUIDOS (CLP2)

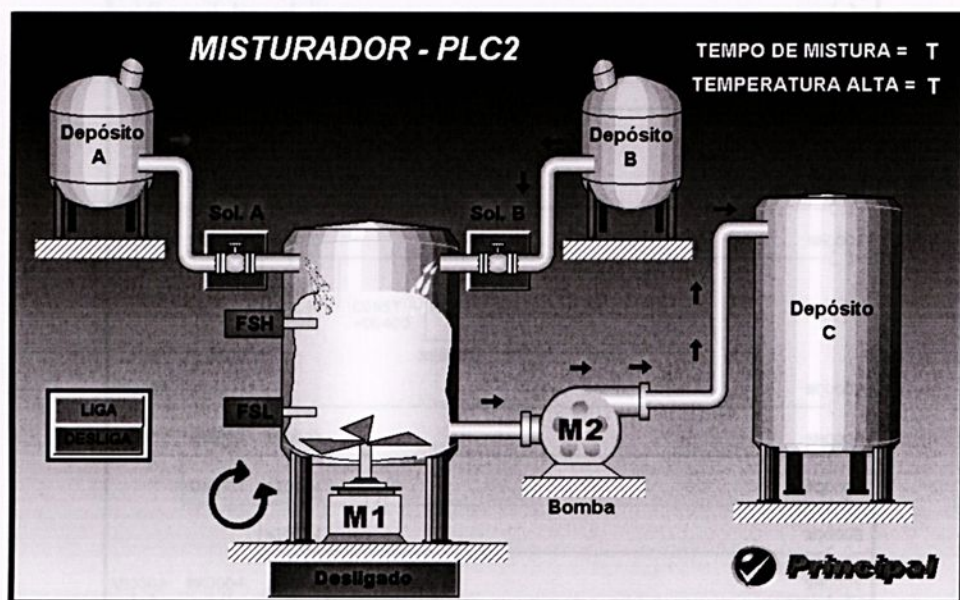


Figura 5.5 - tela do software supervisório para o misturador de líquidos.

A etapa mais importante do processo é descrita a seguir. Ela é responsável pela fabricação do produto final, chamado aqui de líquido C.

Basicamente, este processo funciona da seguinte forma: dando-se o comando liga, as válvulas solenóides abrem, permitindo que os líquidos A e B sejam depositados no reservatório de mistura. Quando o sensor FSH indicar nível alto, as válvulas se fecham e o motor M1 entra em funcionamento, misturando os dois líquidos até se tornarem homogêneos. Para realizar essa mistura, esse motor gira em dois sentidos, com tempos pré-determinados na lógica programada. Ao terminar o tempo necessário para alcançar a homogeneidade, o motor M1 pára de funcionar e M2 é acionado, transferindo o líquido misturado para o depósito C.

Assim, o nível do líquido, do reservatório de mistura, abaixa até atuar no sensor de nível baixo FSL; esse efetuará, através da lógica programada em linguagem *Ladder*, o reinício de todo o processo de mistura, reabrindo as válvulas solenóides.

As duas próximas figuras descrevem a lógica programada, através da linguagem *Ladder*. Este processo e sua lógica de controle também são detalhadamente explicados no apêndice B desta monografia, onde são descritos os novos comandos e procedimentos para inserir tal processo na rede e para acrescentar a ele a etapa de aquecimento do líquido C, através de um controlador PID, antes de sua transferência para o depósito C.

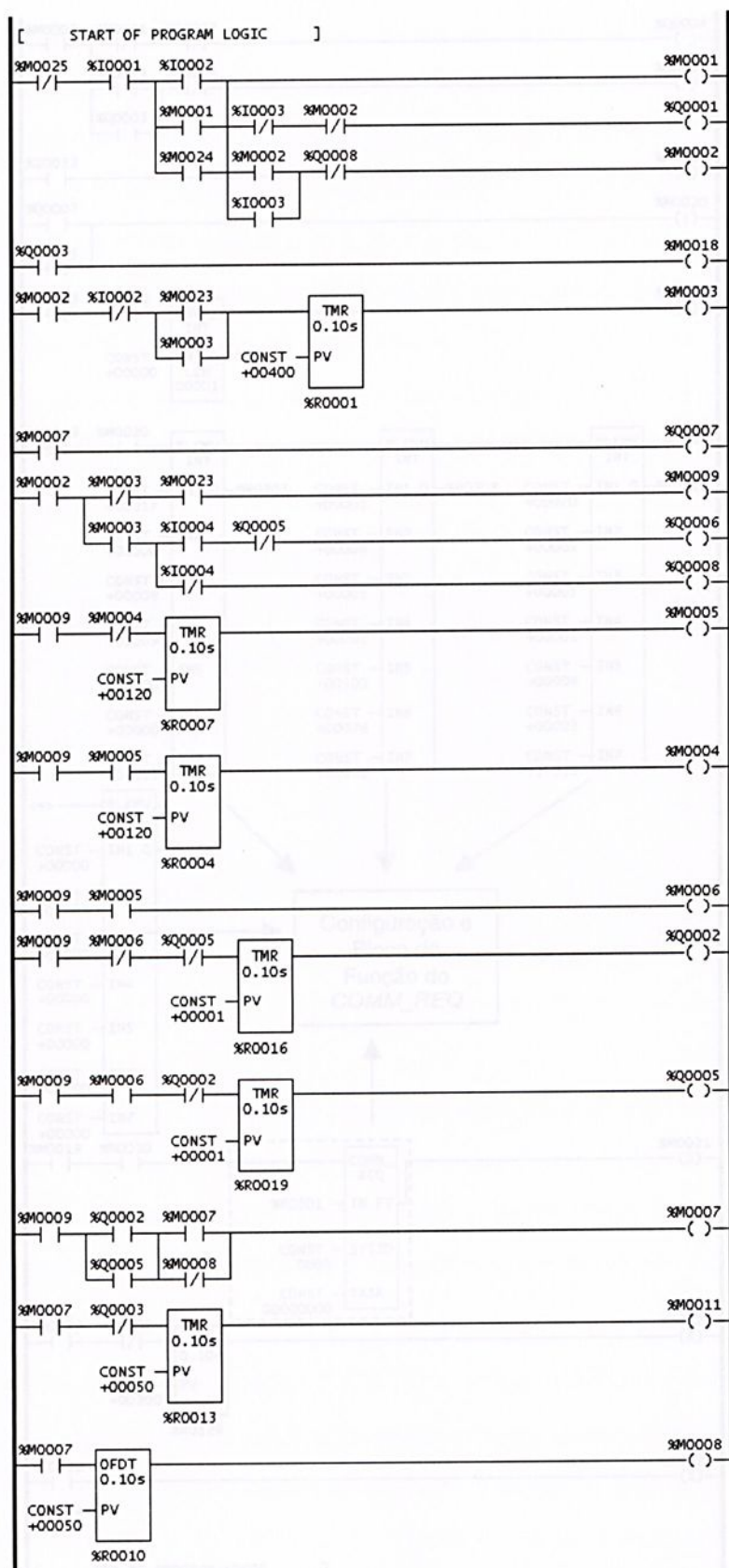


Figura 5.6 - primeira parte da lógica de programação Ladder para o misturador.

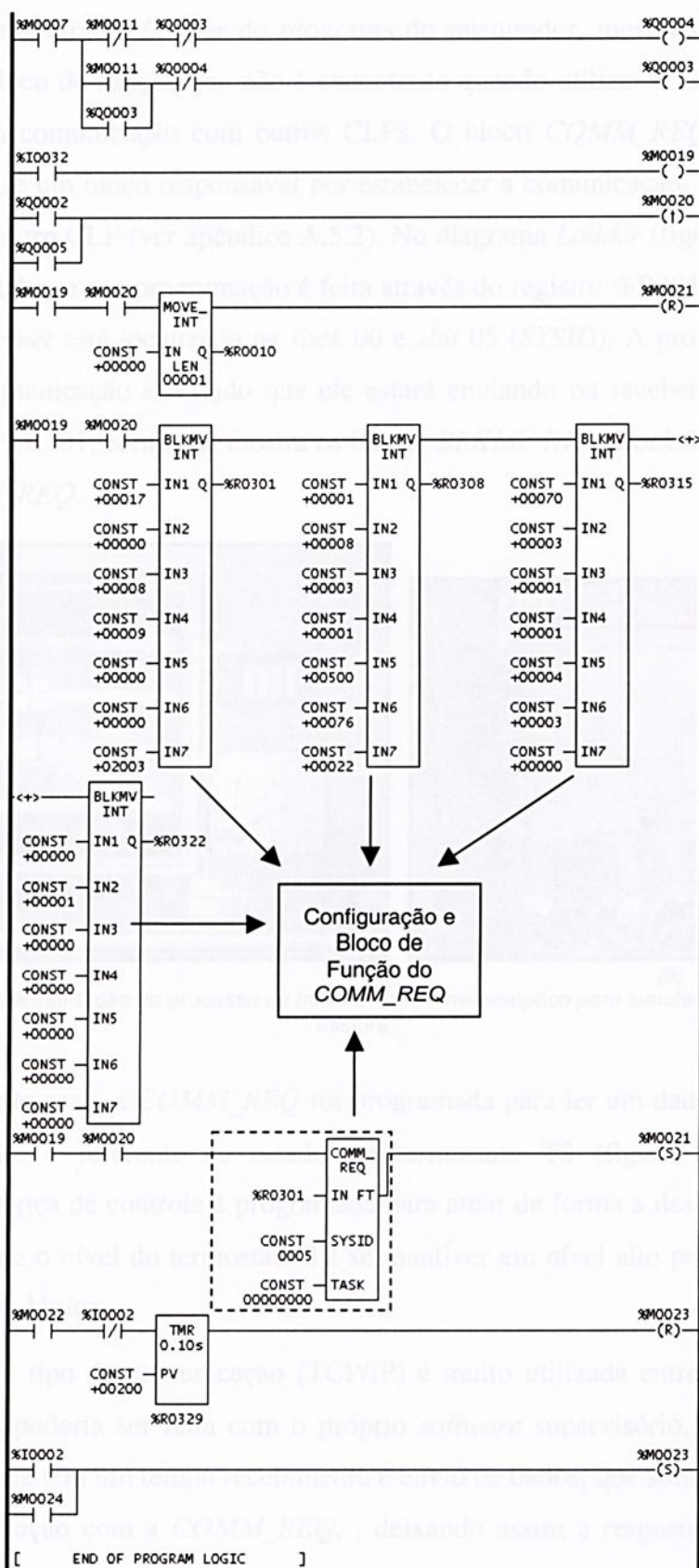


Figura 5.7 - segunda parte da lógica de programação Ladder para o misturador.

5.3 SISTEMA No diagrama *Ladder* do programa do misturador, mostrado nas figuras 5.6 e 5.7, é visto um bloco de função que não é encontrado quando utiliza-se CLPs que controlam um processo sem comunicação com outros CLPs. O bloco *COMM_REQ*, em destaque no diagrama *Ladder*, é um bloco responsável por estabelecer a comunicação, através de TCP/IP, de um CLP com outro CLP (ver apêndice A.5.2). No diagrama *Ladder* (figura 5.7), esse bloco está configurado tal que sua programação é feita através do registro %R301 (IN) e ele informa que o cartão *Ethernet* está localizado na *rack* 00 e *slot* 05 (SYSID). A programação de como será feita esta comunicação e o dado que ele estará enviando ou recebendo é feita então a partir do registro %R301, conforme mostra os blocos *BLKMV INT* (*Block Movement*), na *rung* anterior a *COMM_REQ*.

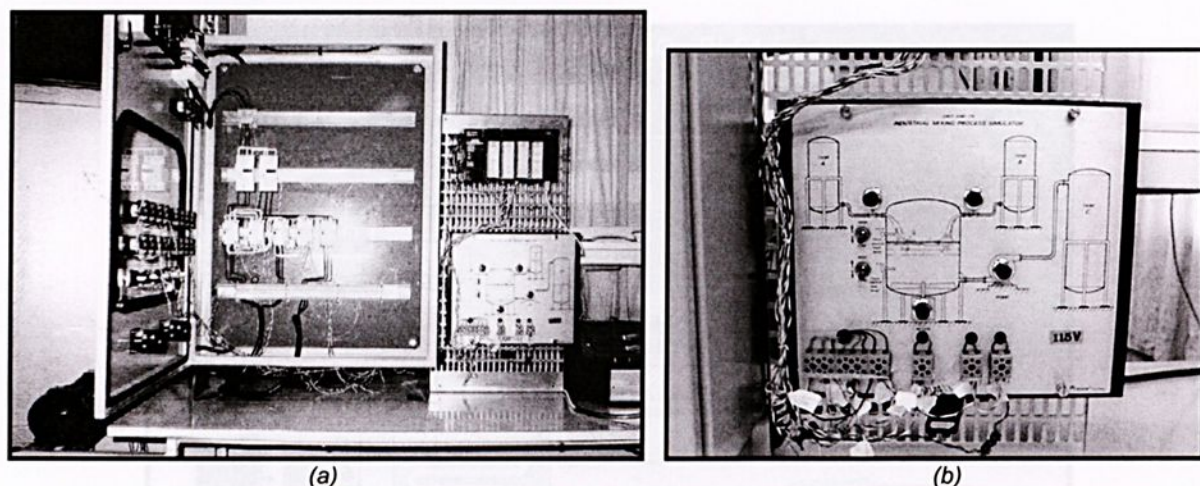


Figura 5.8 - (a) implementação do processo de mistura, (b) painel sinóptico para simulação do processo de mistura.

Neste caso, a *COMM_REQ* foi programada para ler um dado de outro CLP. O dado, em questão, é referente ao estado do termostato T2 (figura 5.9 – sistema de refrigeração). A lógica de controle é programada para atuar de forma a desligar o processo de mistura sempre que o nível do termostato T2 se mantiver em nível alto por um tempo maior que o pré-fixado na lógica.

Este tipo de comunicação (TCP/IP) é muito utilizada entre CLPs, sendo que esta comunicação poderia ser feita com o próprio *software* supervisor, que usa o mesmo protocolo, porém ele tem um tempo recebimento e envio de dados, que somados, é superior ao tempo da comunicação com a *COMM_REQ*, deixando assim a resposta do controle mais lenta.

5.3 SISTEMA DE REFRIGERAÇÃO (CLP1)

Dentre os três processos automatizados, da simulação industrial, esse é o mais complexo. Uma vez o sistema de refrigeração ligado, os motores M1 e M2 entram em funcionamento. O motor M1 leva a água gelada (objeto do controle desse sistema) para o local desejado para ser refrigerado, enquanto que o motor M2 é usado para bombear a água do condensador, permitindo que esta água permaneça a temperatura ambiente, ou próxima dela. O ventilador da torre entrará em funcionamento logo a seguir, sendo utilizado para ajudar no processo de resfriamento da água do condensador; esse ventilador trabalha em duas velocidades, controladas por um inversor de frequência (ver apêndice B.3.2), dependendo da temperatura lida no termostato T1.

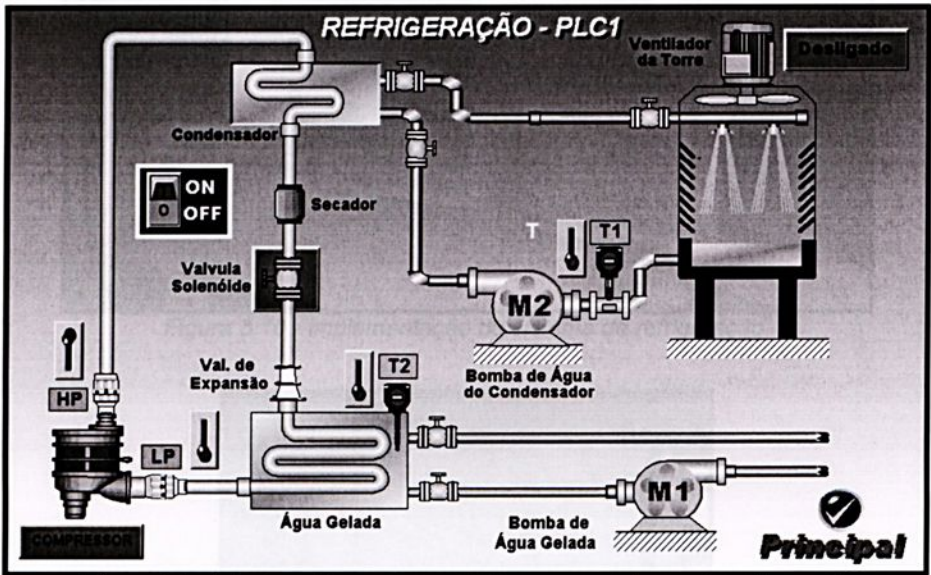


Figura 5.9 - tela do software supervisor para o sistema de refrigeração

O compressor entrará em funcionamento sempre que o termostato T2 acusar temperatura alta para o reservatório de água gelada. Quando o compressor entrar em funcionamento, permitirá o fluxo do gás em alta pressão e temperatura até o condensador, que transformará esse gás em líquido a alta pressão. Esse líquido em alta pressão passará por uma válvula de expansão que transformará o líquido novamente em gás, porém em baixa temperatura, fazendo com que as serpentinas, contidas no reservatório de água geladas, gelem e, através da troca de calor também gele a água contida no reservatório de água gelada.

Esta água gelada vai ser transportada até o equipamento a ser refrigerado. Nesta simulação industrial, esta água gelada está refrigerando o motor M1 do processo de mistura,

pois este motor é ligado e desligado várias vezes, provocando um aumento em sua temperatura, podendo causar problema em seus enrolamentos.

O pressostato HP é utilizado por segurança, impedindo que o compressor funcione quando a pressão for muito alta, e o pressostato LP é usado para controlar a pressão do gás na entrada do compressor, controlando assim o momento correto para desligar o compressor.

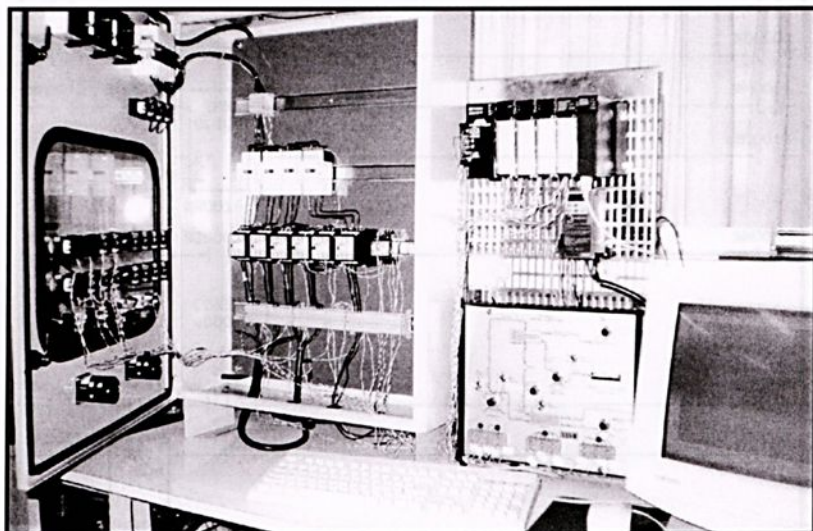


Figura 5.10 - implementação do sistema de refrigeração.

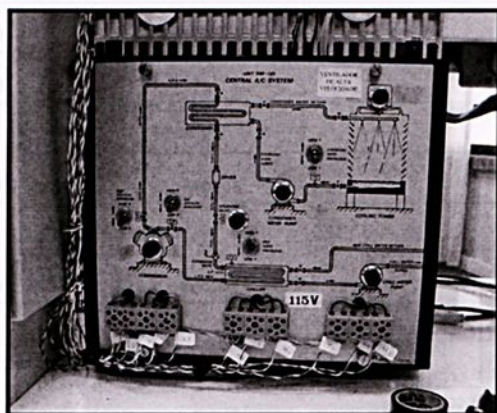


Figura 5.11 - Painei sinóptico para simulação do sistema de refrigeração.

O CLP1, é responsável pelo controle do sistema de refrigeração e de sinalização via pêndulo invertido (apêndice B.3.3), que utiliza sinal analógico, porém o sistema de sinalização junto com os comandos de intertravamento em rede de comunicação TCP/IP, não são apresentados neste capítulo; para maiores detalhes do processo completo (diagramas elétrico e *Ladder*) ver o apêndice B.

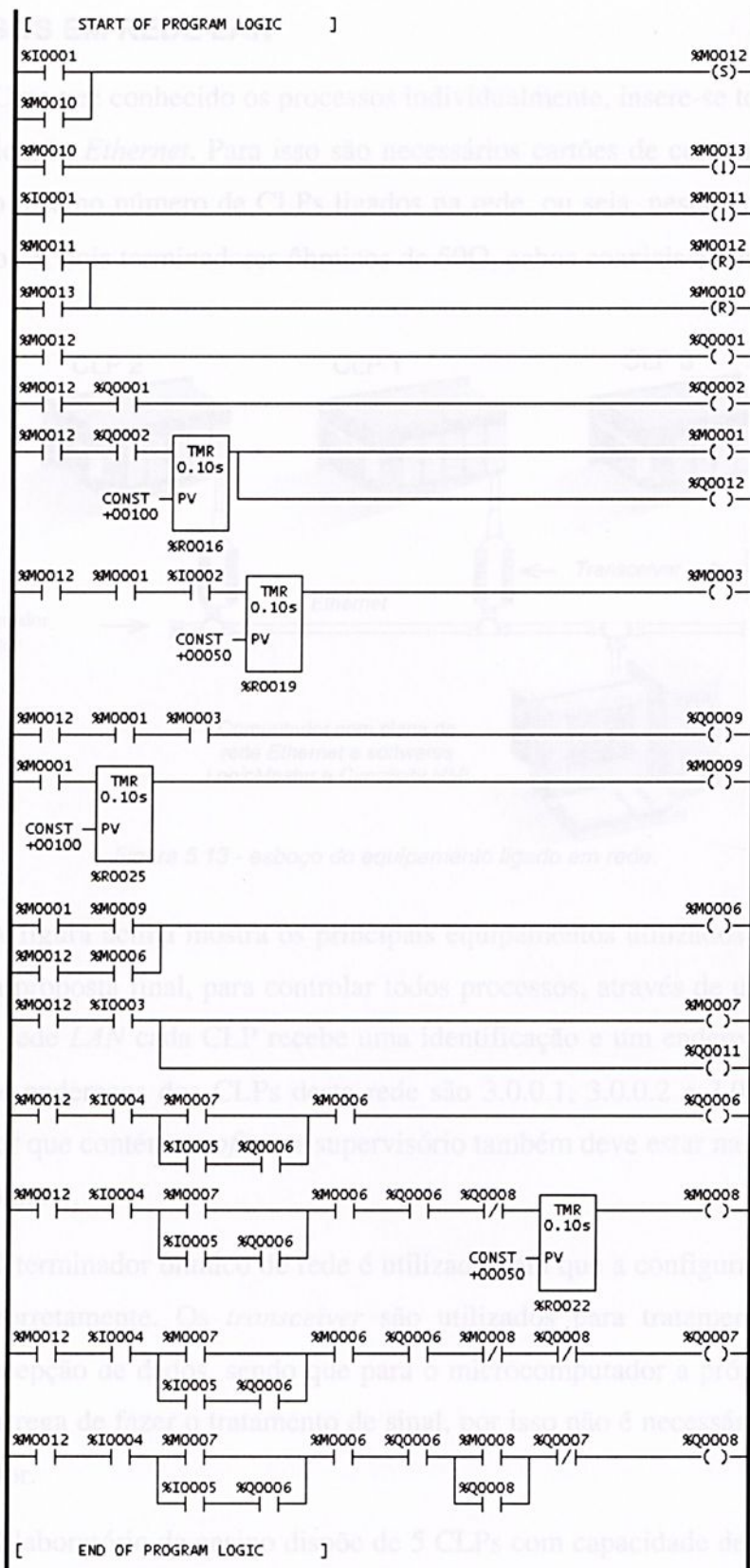


Figura 5.12 - lógica de programação Ladder para o sistema de refrigeração.

5.4 PROCESSOS EM REDE LAN

Uma vez conhecido os processos individualmente, insere-se todos numa mesma rede de comunicação *Ethernet*. Para isso são necessários cartões de comunicação *Ethernet* e *transceivers*, no mesmo número de CLPs ligados na rede, ou seja, nesta simulação industrial serão necessários 3; dois terminadores ôhmicos de 50Ω, cabos coaxiais e conectores-T do tipo BNC.

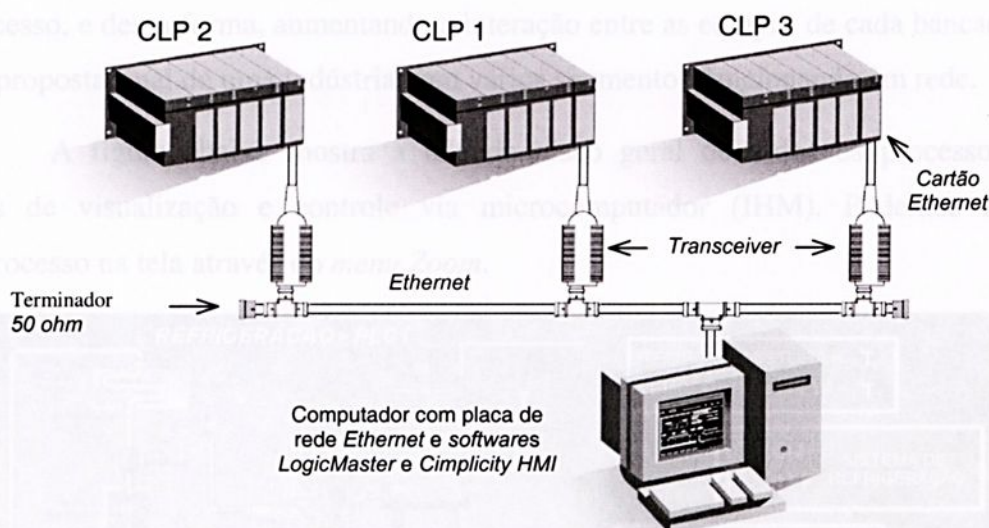


Figura 5.13 - esboço do equipamento ligado em rede.

A figura acima mostra os principais equipamentos utilizados na pequena rede, da simulação da proposta final, para controlar todos processos, através de um sistema central de controle. Na rede *LAN* cada CLP recebe uma identificação e um endereço. Uma sugestão para escolha dos endereços dos CLPs desta rede são 3.0.0.1, 3.0.0.2 e 3.0.0.3, sendo que o microcomputador que contém o *software* supervisor também deve estar na mesma rede, com sugestão de *IP* 3.0.0.4.

O terminador ôhmico de rede é utilizado para que a configuração, na topologia *bus*, funcione corretamente. Os *transceiver* são utilizados para tratamento dos sinais de transmissão e recepção de dados, sendo que para o microcomputador a própria placa de rede *Ethernet* se encarrega de fazer o tratamento de sinal, por isso não é necessário um *transceiver* para o computador.

O laboratório de ensino dispõe de 5 CLPs com capacidade de comunicação em rede, porém como possui apenas 3 *transceivers* não pode-se colocar todos CLPs funcionando

em rede, por isso a sugestão da proposta final é feita com apenas 3 CLPs funcionando em rede.

Sugere-se então a aquisição de mais dois *transceivers* para a que a disciplina Prática em Controle e Acionamentos de Máquinas Elétricas em Processos Industriais possa ser ministrada com capacidade de colocar todas as bancadas ligadas em rede, ficando um microcomputador, com o *software* supervisor, responsável pelo controle e supervisão de todo o processo, e dessa forma, aumentando a interação entre as equipes de cada bancada para finalizar a proposta final de uma indústria com vários segmentos funcionando em rede.

A figura abaixo mostra a tela de visão geral de todos os processos e as alternativas de visualização e controle via microcomputador (IHM). Podemos ampliar qualquer processo na tela através do *menu Zoom*.

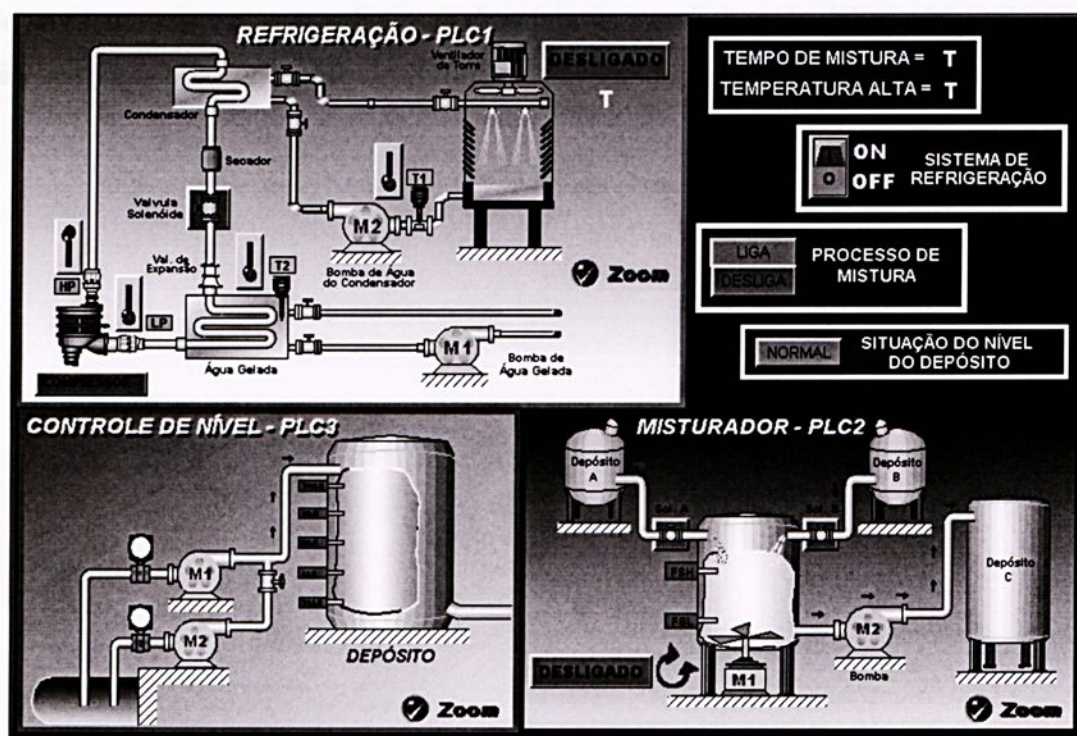


Figura 5.14 - tela de visão geral dos processos.

Todos os processos são supervisionados na tela do computador, sendo que o usuário poderá atuar nos principais pontos da rede de forma remota, bastando um "click" com o *mouse* para ligar e desligar o processo. Outra importante função de um sistema operando em rede é que ele pode ser controlado via *Web*, ou seja, o operador pode, em qualquer lugar,

acessando a *Web* intervir no sistema. Para isso são necessários endereços (*IP*) de rede corporativa, e da licença do *software Web View* da *GeFanuc*.

Os detalhes de como confeccionar uma tela de supervisão e controle de processos será apresentado no apêndice C desta monografia, onde mostra-se em detalhes como foi confeccionado a tela de supervisão do processo de mistura e os seus pontos controlados. No apêndice B são mostradas também as telas referentes à etapa de aquecimento do líquido C e de supervisão via pêndulo invertido.

Com a finalização do projeto final e funcionamento correto de toda a rede (intertravamentos, supervisão e controle), os alunos estarão aptos e altamente qualificados para atuarem no ramo de automação industrial, pois receberam um conhecimento bastante abrangente sobre o assunto e, graças a atualização tecnológica do laboratório e ao ambiente criado para aproximar as condições industriais às condições do laboratório, eles também estarão com uma boa concepção dos problemas que poderão encontrar no campo.



CONCLUSÃO E OBSERVAÇÕES FINAIS

Ficou bastante claro que a pesquisa e o ensino em automação e processos de controle no curso de Engenharia Elétrica da FEIS tem evoluído bastante em curto espaço de tempo, isto porque entendemos que há necessidade de oferecer essa formação, visto as necessidades e exigências do mercado, não apenas brasileiro, mas mundial. É claro que ainda há muito para aprender e pesquisar, mas estamos evoluindo no sentido de nos ajustar em relação às novas tendências do mercado tecnológico, onde a busca por soluções de baixo custo com alta produtividade e qualidade, vem permitindo que soluções cada vez mais interessantes sejam encontradas. E com toda esta situação, temos certeza que o empenho dos profissionais, alunos e pesquisadores em automação e controle, no campus da FEIS/UNESP, ainda mostrarão novas perspectivas e soluções para esta área.

A nova disciplina “*Prática em Controle e Acionamentos de Máquinas Elétricas em Processos Industriais*” já é um resultado prático de nosso trabalho e tem caminhado paralelamente à pesquisa, ou seja, conforme novas etapas são abertas pelos pesquisadores, os conhecimentos adquiridos são inseridos no curso, no intuito de promover a melhor formação possível aos alunos. Por isso propomos ter sempre alunos pesquisadores e profissionais, envolvidos

A procura de alunos interessados em pesquisa nesta área tem aumentado bastante, por isso e por causa da constante evolução tecnológica em automação, necessitaremos de novos equipamentos, não para substituir os atuais, pois estamos bem servidos de equipamentos, mas para somar a eles e oferecer aos nossos alunos e pesquisadores uma melhor condição de trabalho.

Outro fator que mostra o quanto o nosso trabalho tem despertado o interesse no meio acadêmico, são os convites que o grupo de pesquisa está recebendo para oferecer palestras, apresentar trabalhos e promover cursos para alunos não matriculados nesta



disciplina e para alunos de escolas técnica, consolidando a importância do ensino em automação e nos conduzindo a estar sempre atentos às novas tendências do mercado tecnológico. Desta forma, continuaremos os trabalhos de pesquisa na área, buscando sempre aprimorar os conhecimentos, proporcionando ao curso de Engenharia Elétrica da Faculdade de Engenharia de Ilha Solteira (FEIS/UNESP) um ensino de alto nível. Dentre os tópicos que ainda podem ser aprofundados na pesquisa, um que pode despertar muito interesse é a utilização da automação de processos utilizando a *Web* e a comunicação via satélite como meio de transmissão de dados.

Este trabalho e os seus resultados, motivou-nos a buscar “novos horizontes” de aplicação da automação, através da pesquisa às novas necessidades do mercado. E como resultado desta, estaremos numa nova etapa, dedicando nosso trabalho à pesquisa, junto à comunidade científica e tecnológica, buscando contribuir às necessidades encontradas, principalmente no Brasil, em controle de recursos naturais do setor petrolífero e de gás natural.

[1] ELETRÔNICA, revista, *Tecnologia, Informática e Automação*; Ed. Saber Ltda., São Paulo, nº 339, abril 2001.

[16] ELETRÔNICA, revista, *Tecnologia, Informática e Automação - Especial*; Ed. Saber Ltda., São Paulo, nº 2, fevereiro 2001.

[17] ELETRÔNICA, revista, *Tecnologia, Informática e Automação*; Ed. Saber Ltda., São Paulo, nº 334, novembro 2000.

[18] TOSHIBA CORPORATION; “User’s Manual - Basic Hardware and Function”.

[19] GE FANUC AUTOMATION; “Programmable Control Products”; junho 1999.

[10] CONTROLE & INSTRUMENTAÇÃO, revista; Edição nº 50, setembro de 2000.

[21] CONTROLE & INSTRUMENTAÇÃO, revista; Ed. 50 nº 48, julho de 2000.

[12] CONTROLE & INSTRUMENTAÇÃO, revista; Edição nº 47, junho de 2000.

[13] VEJA, revista; Ed. 1061, 9 de agosto de 2000.

[14] DEGEM SYSTEMS; “Práticas de Laboratório en Arranque y Control de Motores Eléctricos”; curso ICS-100 “Construcción y Control Industrial”.

[15] MULTIQ™ QUANSER CONSULTING; “Programming Manual”.

[16] FUJI ELECTRIC COE; “Instruction Manual FVR-C/S-MUX Drive Series”.



BIBLIOGRAFIA

- [1] NATALI, Ferdinando; *Automação Industrial*; Ed. Érica, São Paulo, 2000.
- [2] MAGRINI, Luiz Carlos; *Integração de Subsistemas Para Compor a Automação da Distribuição de Energia Elétrica*; Tese de Doutorado, USP, São Paulo, 1999.
- [3] MIYAGI, Paulo Eigi; *Controle Programável*; Ed. Edgard Blücher Ltda, São Paulo, 1997.
- [4] ELETRÔNICA, revista; *Tecnologia, Informática e Automação*; Ed. Saber Ltda., São Paulo, nº 341, junho 2001.
- [5] ELETRÔNICA, revista; *Tecnologia, Informática e Automação*; Ed. Saber Ltda., São Paulo, nº 339, abril 2001.
- [6] ELETRÔNICA, revista; *Tecnologia, Informática e Automação - Especial*; Ed. Saber Ltda., São Paulo, nº 2, fevereiro 2001.
- [7] ELETRÔNICA, revista; *Tecnologia, Informática e Automação*; Ed. Saber Ltda., São Paulo, nº 334, novembro 2000.
- [8] TOSHIBA CORPORATION; *"User's Manual – Basic Hardware and Function"*.
- [9] GE FANUC AUTOMATION; *"Programmable Control Products"*; junho 1999.
- [10] CONTROLE & INSTRUMENTAÇÃO, revista; Edição nº 50, setembro de 2000.
- [11] CONTROLE & INSTRUMENTAÇÃO, revista; Edição nº 48, julho de 2000.
- [12] CONTROLE & INSTRUMENTAÇÃO, revista; Edição nº 47, junho de 2000.
- [13] VEJA, revista; Ed. 1661, 9 de agosto de 2000.
- [14] DEGEM SYSTEMS; *"Práticas de Laboratório en Arranque y Control de Motores Electricos"*; Curso ICS-100 "Commutacion y Control Industrial".
- [15] MULTIQTM QUANSER CONSULTING; *"Programming Manual"*;
- [16] FUJI ELECTRIC/GE; *"Instruction Manual FVR-C9S-2UX Drive Series"*.



SOFTWARE DE PROGRAMAÇÃO: LOGICMASTER
TCP/IP V9.02

A.1 APRESENTAÇÃO DO SOFTWARE

Existem, no mercado, vários *softwares* de programação de CLPs, quase todos com tecnologia proprietária, ou seja, produzidos para serem usados somente por um determinado fabricante. Apresenta-se aqui, o *software* *LogicMaster TCP/IP V9.02*, da GE Fanuc, fornecido junto ao equipamento do laboratório de ensino em automação, adquirido através de verbas dos programas REENGE e PROIN. Descreve-se, neste apêndice, algumas de suas ferramentas, bem como a configuração necessária para funcionar em rede de comunicação Ethernet.

APÊNDICE A

Software de programação: LogicMaster TCP/IP V9.02

Após instalação do software, é criado, no item *menu* *Iniciar/Programas/PLC Programmer*, um ícone de atalho, com o qual se abre a tela principal do *software* (figura A.1). Será exposta a seguir, a utilidade dos principais *menus* do *software* de programação e configuração.

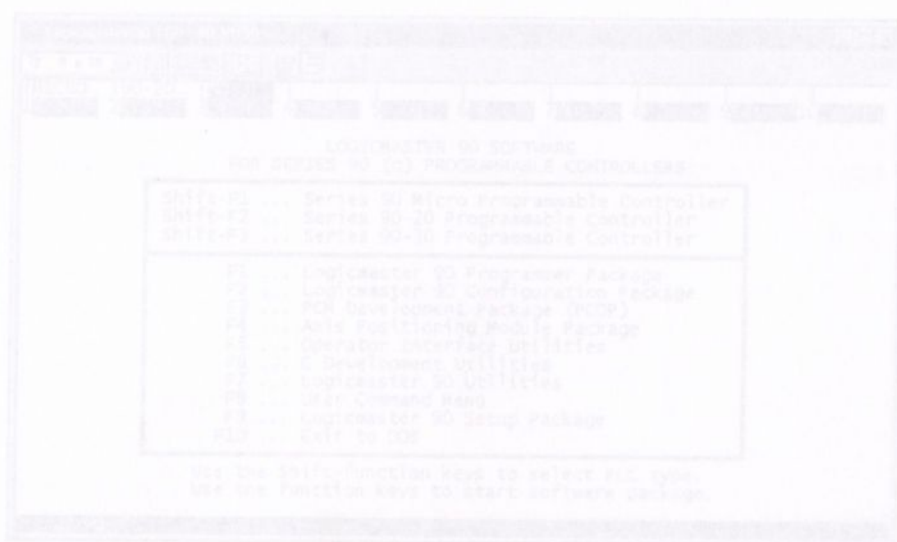


Figura A.1-janela principal do software LogicMaster TCP/IP v9.02

A figura A.2 mostra os *menus* da janela principal do *software* de programação. Para escolher qualquer *menu* da linha inferior basta pressionar a tecla *F1*, *F2*,... ou *F10*.

APÊNDICE A

SOFTWARE DE PROGRAMAÇÃO: LOGICMASTER TCP/IP V9.02

A.1 APRESENTAÇÃO DO SOFTWARE

Existem, no mercado, vários *softwares* de programação de CLPs, quase todos com tecnologia proprietária, ou seja, produzidos para serem usados somente por um determinado fabricante. Apresenta-se aqui, o *software LogicMaster TCP/IP V9.02*, da *GE Fanuc*, fornecido junto ao equipamento do laboratório de ensino em automação, adquirido através de verbas dos programas REENGE e PROIN. Descreve-se, neste apêndice, algumas de suas ferramentas, bem como a maneira correta de configurá-lo para funcionar em rede de comunicação *Ethernet*.

Após instalação do *software*, é criado, no item *menu Iniciar/Programas/PLC Programmer*, um ícone de atalho, com o qual se abre a tela principal do *software* (figura A.1). Será exposta a seguir, a utilidade dos principais *menus* do *software* de programação e configuração.

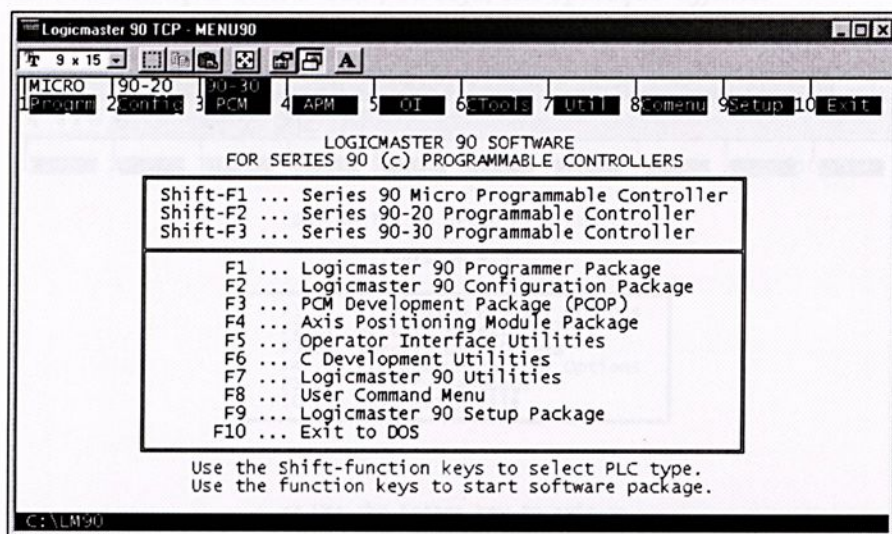


Figura A.1- janela principal do software LogicMaster TCP/IP v9.01

A figura A.2 mostra os *menus* da janela principal do *software* de programação. Para escolher qualquer *menu* da linha inferior basta pressionar a tecla *F1*, *F2*,... ou *F10*,

correspondente ao comando desejado; por exemplo, para escolher o *menu Config* pressiona-se a tecla *F2*. Para escolher o *menu* da linha superior basta pressionar a tecla *Shift+(F1, F2 ou F3)*; por exemplo, para escolher o *menu 90-30* pressiona-se as teclas *Shift+F3*.



Figura A.2 - menu de atalhos para os comandos da janela principal

Utiliza-se aqui o modelo de CLP *GE Fanuc* série 90-30, pois é o modelo existente no laboratório de ensino da FEIS, logo deve-se escolher o *menu 90-30*. Os *menus* mais importantes da tela principal e suas principais funções são:

- **Progrm:** utilizado para entrar no *menu* de edição do programa (*F1*);
- **Config:** usado para configurar o CLP e seus módulos (*F2*);
- **Útil:** usado para configurar a rede de CLPs (*F7*);
- **Setup:** usado para especificar o tipo de comunicação com o CLP e o padrão de cor de apresentação do *software LogicMaster TCP/IP* (*F9*), e
- **Exit:** *menu* de atalho para sair do *software* (*F10*).

Sem entrar em muitos detalhes, será explicada a função de cada *menu* de programação. Mas antes de iniciar a explicação é necessário verificar se o *software* está configurado para comunicação serial com o CLP. Pressiona-se a tecla *F9*, abrindo o *menu Setup*, e então escolhe-se o *menu comm* (*F4*) para configurar a comunicação serial. Uma observação importante é que todo procedimento explicado neste item é feito inicialmente com o *software* sem comunicação com o CLP, ou seja, em operação *Offline*.

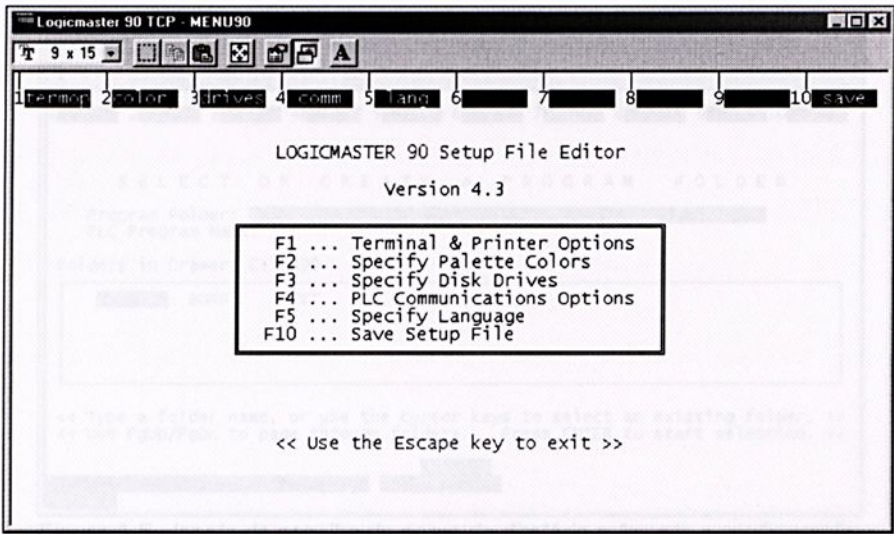


Figura A.3 - janela referente ao menu Setup.

Usando-se a tecla *Tab* muda-se o tipo de comunicação até encontrar a opção *serial port*, conforme figura a seguir.

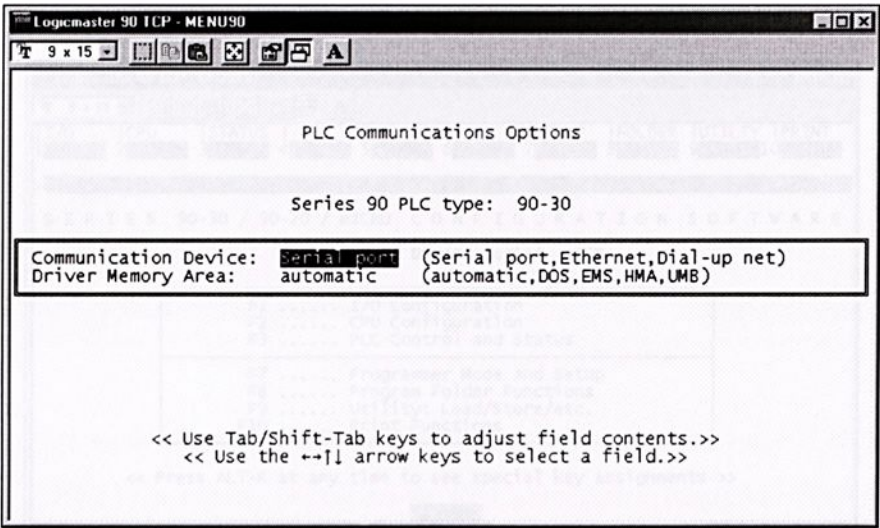


Figura A.4 - janela de seleção da comunicação do CLP.

Pressiona-se tecla *Esc* e volta-se a janela do *menu Setup*, onde pressionando a tecla *F10*, é salva a configuração para o tipo de comunicação serial. Feito isso, pressiona-se a tecla *Esc* outras duas vezes, voltando à janela principal do programa.

A.2 MENU CONFIG

Este deve ser o primeiro *menu* a ser explanado, visto que nele são configurados o CLP e a localização todos os seus módulos. Assim, pressionando a tecla *F2*, abri-se a tela do *menu* de configuração.

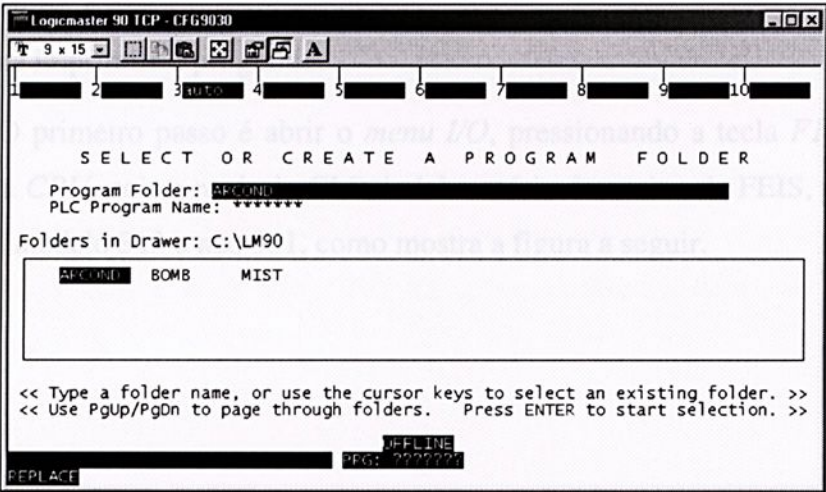


Figura A.5 - janela de escolha do nome do diretório referente a configuração.



Nesta janela escolhe-se um nome para o *config* que poderá ser o mesmo nome que será utilizado no programa. Se o nome escolhido não existe ainda, o programa perguntará se deseja criar um diretório com este nome e a resposta deve ser sim (*yes*).

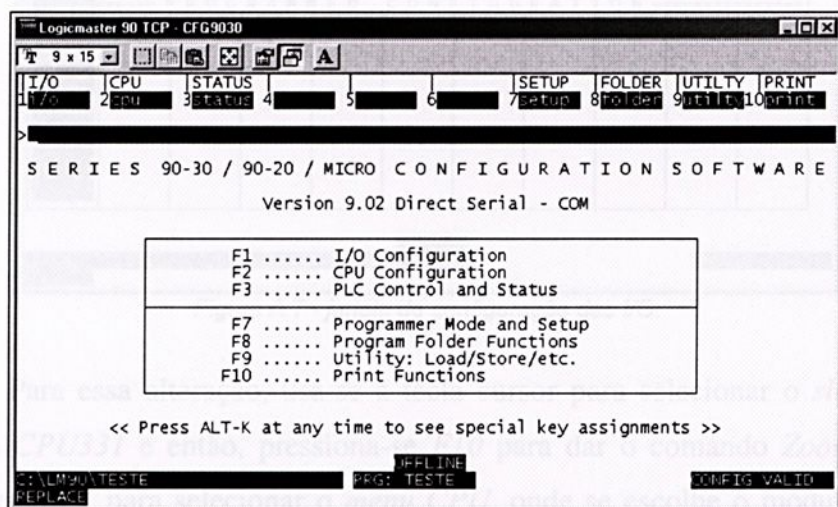


Figura A.6 - janela de configuração.

Nesta nova janela aparecem outros *menus*, cujas principais funções são:

- **I/O:** configurar os módulos de entrada e saída do CLP;
- **CPU:** mudar o *ID* do CLP que está em operação *Online*;
- **STATUS:** verificar falhas de comunicação e nos módulos, quando estiver em comunicação *Online*;
- **SETUP:** selecionar o CLP que estará em comunicação com o *software*, quando existir mais de um CLP ligado em rede;
- **FOLDER:** para trabalhar com os arquivos criados na programação;
- **UTILITY:** utilizado para enviar, receber, apagar e comparar dados com o CLP em operação *Online*, e
- **PRINT:** para imprimir o programa.

O primeiro passo é abrir o *menu I/O*, pressionando a tecla *F1*, então deve-se trocar a *rack* e a *CPU*, pois a *rack*, do CLP do laboratório de ensino da FEIS, possui apenas 5 *slots* e a *CPU* é modelo 313 e não 331, como mostra a figura a seguir.

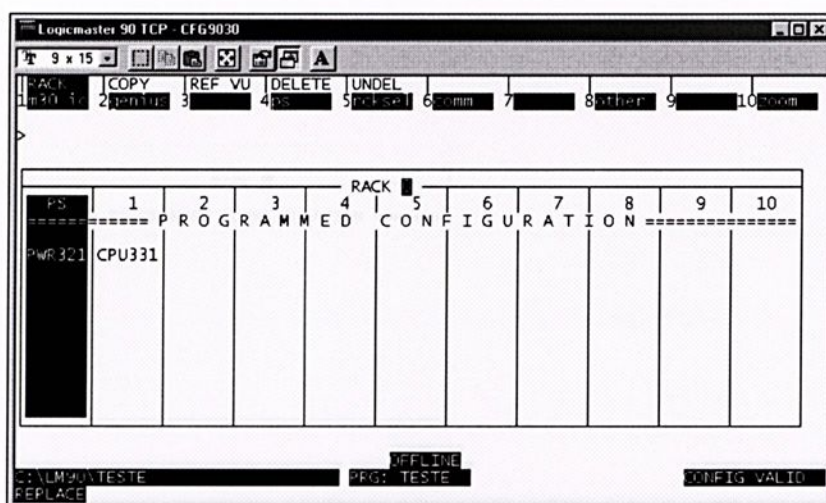


Figura A.7 - janela de configuração das I/O.

Para essa alteração, usa-se a tecla cursor para selecionar o *slot* 1, onde está configurado a *CPU331* e então, pressiona-se *F10* para dar o comando *Zoom*. Na próxima janela use a tecla *F1* para selecionar o *menu CPU*, onde se escolhe o módulo com modelo 313, como mostra a próxima figura.

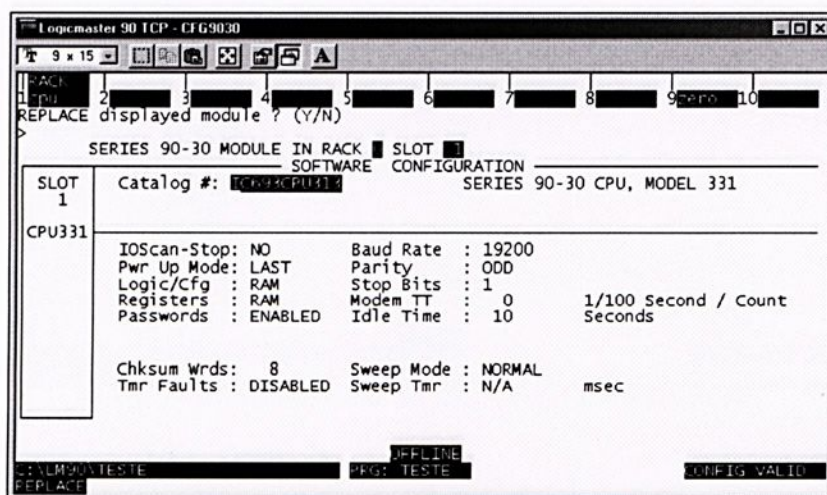


Figura A.8 - tela de seleção da CPU 313.

O *software* perguntará se deseja trocar o módulo antigo pelo novo configurado e a resposta deverá ser sim (Y). Pressionando a tecla *Esc* duas vezes volta-se à tela do *menu I/O*, do *Config*, já com a *rack* e a *CPU* alteradas, conforme *hardware* disponível em laboratório (figura A.9).

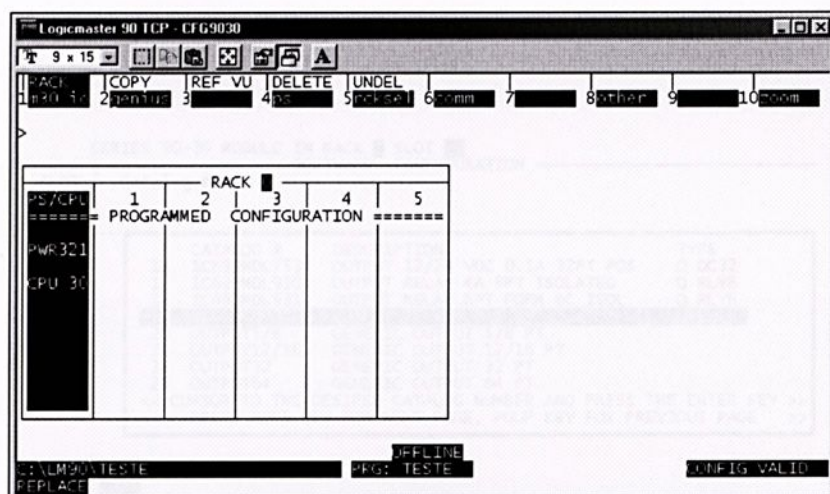


Figura A.9 - janela de configuração das I/O com a rack e a CPU alteradas.

Desta forma já pode-se configurar os módulos presentes em cada *slot*. Como exemplo, será configurado o módulo de saída digital *IC693MDL 940*, no *slot 2*. Para isso, posiciona-se o cursor até o *slot 2* e pressiona a tecla *F1*. Abrirá a janela mostrada na figura A.10, onde deve-se escolher o tipo de módulo que deseja configurar no *slot 2*.

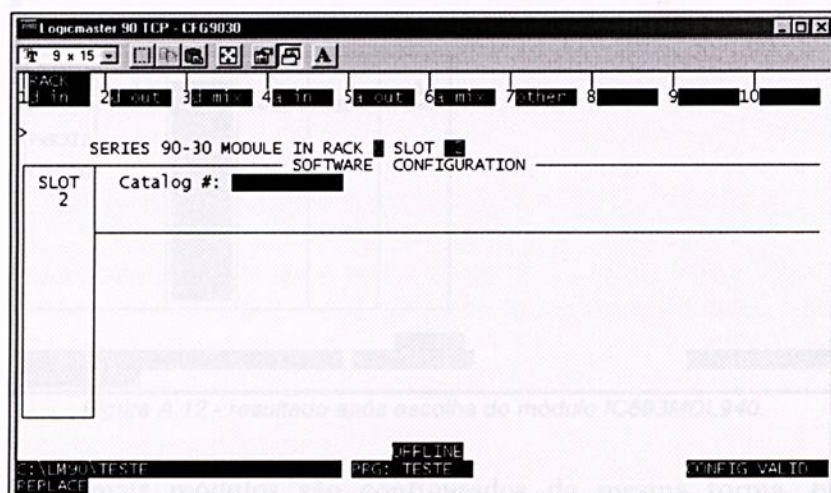


Figura A.10 - janela de escolha dos módulos.

Pressiona-se a tecla *F2*, referente ao *menu d out* (ou seja, saída digital) e então, é escolhido o modelo *IC693MDL940*.

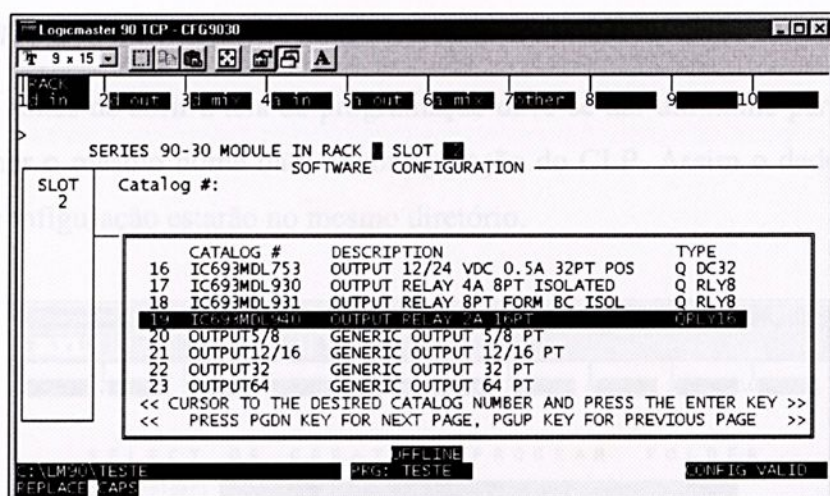


Figura A.11 - escolhendo o módulo IC693MDL940.

Pressionando a tecla *Enter*, é confirmado o módulo, e então, usa-se a tecla *Esc*, para voltar à tela do *menu I/O*, já com o módulo configurado, como mostra a figura a seguir.

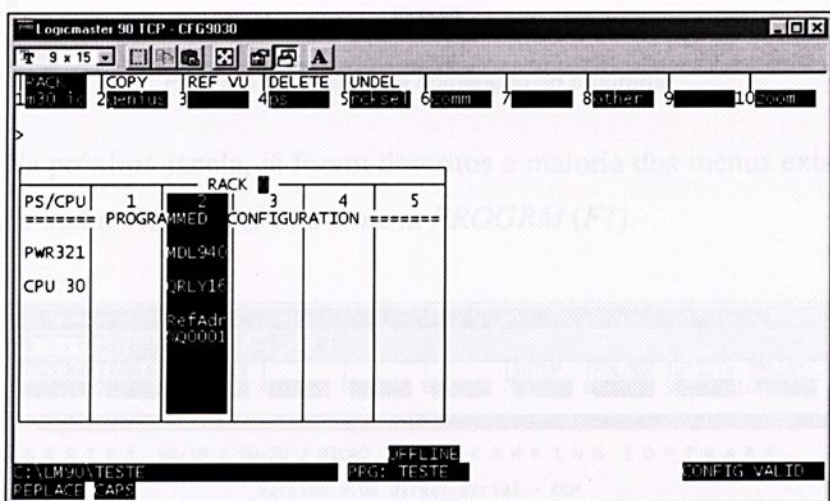


Figura A.12 - resultado após escolha do módulo IC693MDL940.

Os demais módulos são configurados da mesma forma, bastando apenas navegar pelos *menus de hardware* para a escolha do módulo desejado. Para inserir o módulo de comunicação *Ethernet*, escolha o *menu comm (F6)*, dentro do *menu I/O do Config*, e então escolhe-se o *menu ethnet*.

É claro que o *menu Config* tem outras diversas utilidades, mas no momento as informações mais importantes foram descritas; então, deve-se pressionar a tecla *Esc* até voltar a janela principal, onde será escolhido o *menu Program*, pressionando a tecla *F1*.

A.3 MENU PROGRAM

Antes de abrir a tela de programação deve-se dar um nome para o programa, e pode-se escolher o mesmo nome dado à configuração do CLP. Assim o dados referentes ao programa e a configuração estarão no mesmo diretório.

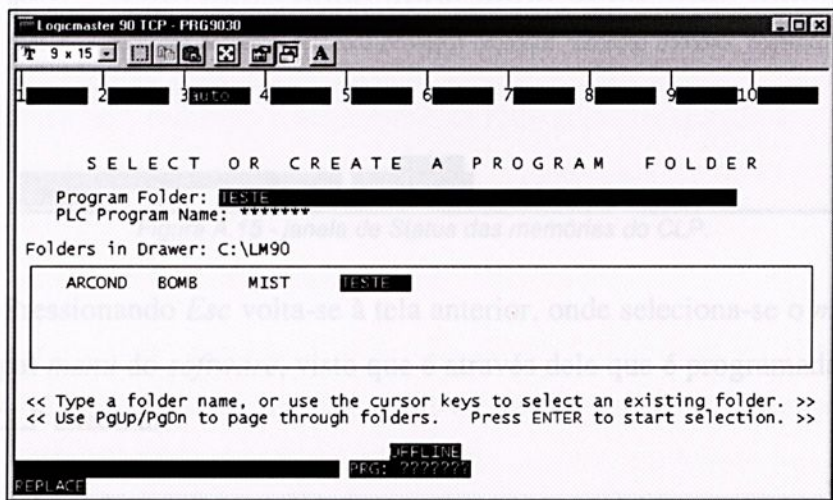


Figura A.13 - janela de nomeação do programa.

Na próxima janela, já foram descritos a maioria dos menus existentes, faltando explicar apenas o *menu TABLES* (F2) e o *menu PROGRM* (F1).

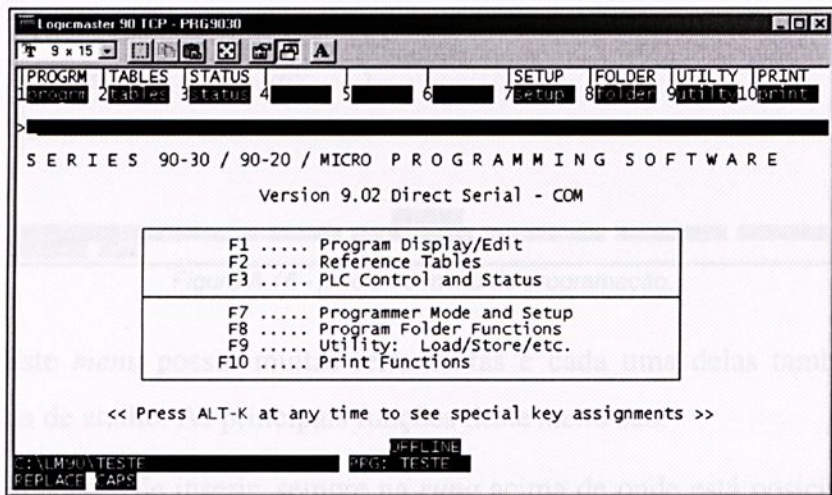


Figura A.14 – menu de programação.

O *menu TABLES*, quando aberto, mostra a situação de todos os locais de memória do CLP. A figura A.15 mostra, por exemplo, todos os locais de memória referente à memória tipo %I, mas pode-se mostrar outros tipos de memória, ou até mesmo ir direto a algum local da memória, bastando dar o endereço desejado.



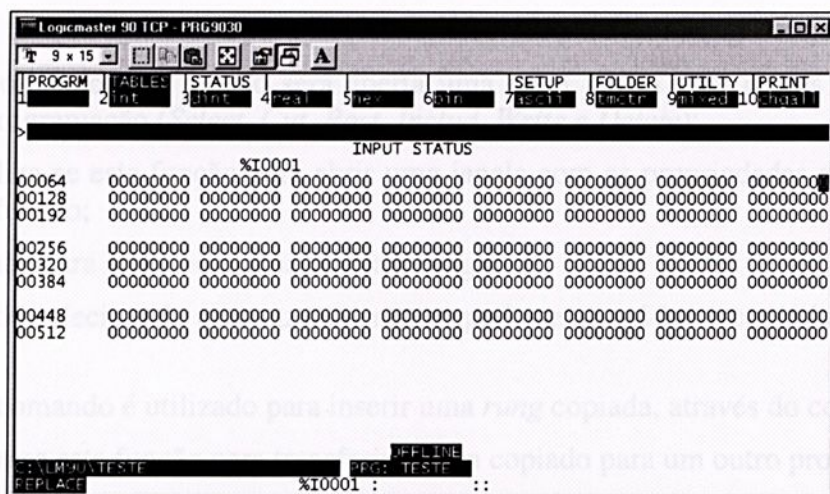


Figura A.15 - janela de Status das memórias do CLP.

Pressionando *Esc* volta-se à tela anterior, onde seleciona-se o *menu PROGRAM*. Este é o principal *menu* do *software*, visto que é através dele que é programada e transferida a lógica para o CLP executar.

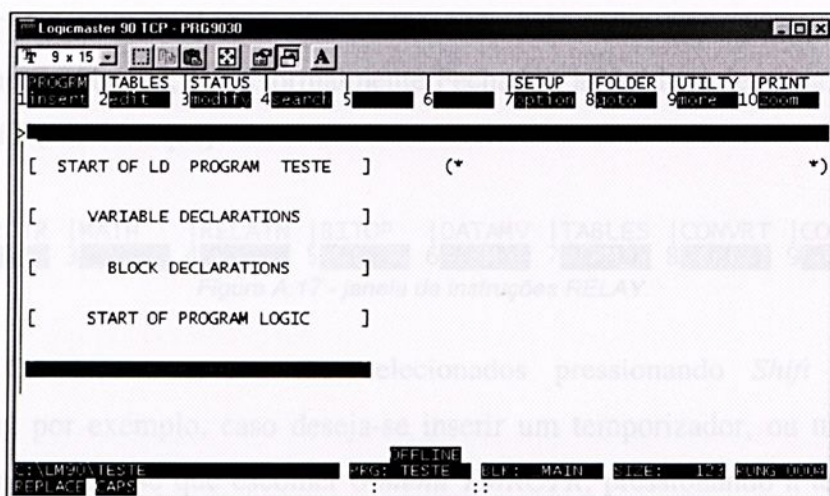


Figura A.16 - janela do menu de programação.

Este *menu* possui muitas ferramentas e cada uma delas também é acessada através das teclas de atalho. As principais funções deste *menu* são:

- **Insert:** sua função é de inserir, sempre na *run*g acima de onde está posicionado o cursor, uma nova *run*g;
- **Edit:** utilizado sempre que deseja-se editar uma linha já programada;
- **Modify:** utiliza-se esta função sempre que se deseja apenas mudar alguma instrução já programada, sem inserir novas funções;
- **Search:** escolhendo este função pode-se localizar o nome dado à alguma instrução, verificando onde está sendo usada;
- **Option:** é usado para configurar o programa para que permita ou não a inserção de mais de uma instrução de saída, usando o mesmo local de memória;

- **Goto:** esta função é utilizada para direcionar o cursor para uma *rung* desejada;
- **More:** escolhendo esta função será aberta uma nova janela com outras ferramentas do *menu* de programação (*Select*, *Cut*, *Past*, *Includ*, *Write* e *Delete*);
- **Zoom:** utiliza-se esta função para abrir uma janela com as propriedades de determinados blocos de função;
- **Select:** usada para selecionar uma ou mais *rungs*;
- **Cut:** quando selecionada uma ou mais *rungs*, pode-se apagá-las e copiá-las, através deste comando;
- **Past:** este comando é utilizado para inserir uma *rung* copiada, através do comando *Cut*;
- **Includ:** usa-se esta função para transferir o item copiado para um outro programa;
- **Write:** é usado para escrever em outro programa a *rung* selecionada, e
- **Delete:** usa-se esta função para apagar a *rung* selecionada.

Para melhor detalhar a utilização destas ferramentas na confecção de uma lógica de controle, é feito a seguir um pequeno exemplo. Primeiro posiciona-se o cursor sobre a última *rung* (*END OF PROGRAM LOGIC*) e então escolhe-se a função *Insert*, pressionando a tecla *F1*; abrirá a nova janela cujas ferramentas são autoexplicativas, mostrando os contatos NA (Normalmente Aberto), NF (Normalmente Fechado), as bobinas de saída, etc., referentes ao *menu RELAY* (em destaque).

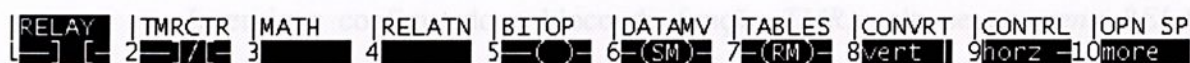
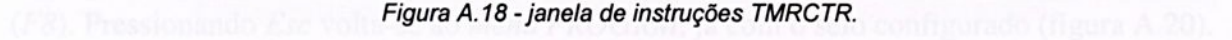


Figura A.17 - janela de instruções RELAY.

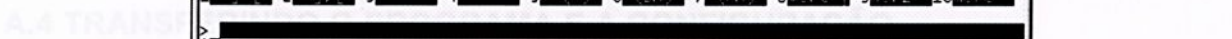
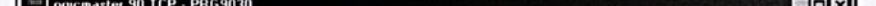
Os demais *menus* são selecionados pressionando *Shift* mais tecla *F* correspondente; por exemplo, caso deseja-se inserir um temporizador, ou um contador, na lógica programada, tem-se que escolher o *menu TMRCTR*, pressionando a tecla *Shift+F2*, e assim por diante.

Na criação da lógica, deste pequeno exemplo, deseja-se inserir, na primeira *rung*, um contato NA, referente ao local de memória %I03, que deverá energizar a bobina de saída %Q01, sempre que estiver em nível lógico "1". Pressiona-se a tecla *F1* e em seguida digita-se, no local onde o cursor pisca, 3i e então confirma-se utilizando a tecla *Enter*; é inserido assim o contato NA referente ao local de memória %I03.

Para inserir a bobina de saída basta pressionar *F5* e então digitar 1q, confirmando através da tecla *Enter*. Pressiona-se novamente a tecla *Enter*, para inserir uma nova *rung*, ou *Esc* para voltar ao *menu Progrm*. Neste exemplo, dá-se continuidade, inserindo uma nova *rung*, cuja função será energizar %M03 cinco segundos após %Q01 ser energizado.



Escolhe-se o temporizador que retarda na energização *tmr*, cuja tecla de atalho é *F3* (figura A.18). A base de tempo do temporizador é de 100 milissegundos, mas pode-se mudar esta base de tempo utilizando a função *tmbase* (*F10*), porém, neste exemplo, utiliza-se a base de tempo 100 milissegundos, fazendo seu registro contar até 50 para obter 5 segundos. Logo, é necessário posicionar o cursor até a entrada *PV* do bloco de função *TMR* e digitar o valor 50. Depois desloca-se o cursor abaixo do bloco, onde existem vários pontos de interrogação, e digita-se somente %R; desta forma, o *software* encontrará sozinho uma sequência de registros que não estão sendo utilizados na programação, para usá-los neste bloco de função (obs.: os blocos de função temporizador e contador utilizam 3 registros para executar suas funções, assim os registros, neste exemplo, %R01, %R02 e %R03 são usados pelo bloco de função *TMR* para contar 5 segundos).



Pressionando *Esc* volta-se ao *menu* *PROGRM*, onde deseja-se agora inserir, em paralelo ao contato NA de %I03, um selo utilizando o contato NA de %M03. Para isso, é

necessário posicionar o cursor sobre a *rung* que deseja-se mudar e então escolher o comando *Edit* (F2), sendo aberto novamente a janela de edição. Movendo o cursor para baixo do contato NA de %I003, pode-se escolher, através da tecla de atalho F1, o novo contato a ser inserido. Dá-se o nome de %M003 para o contato e então pressiona-se *Enter*, que moverá o cursor para direita, onde será necessário apenas inserir linhas verticais, utilizando o *menu vert* (F8). Pressionando *Esc* volta-se ao *menu PROGRAM*, já com o selo configurado (figura A.20).

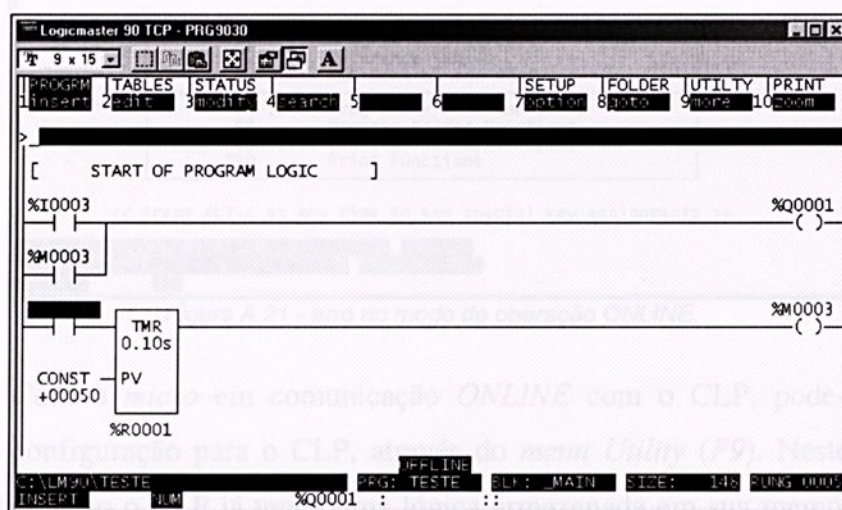


Figura A.20 - inserção do selo sobre o contato NA %I03.

Todas demais funções são utilizadas da mesma forma, não necessitando explicar sobre suas utilizações, ficando o aluno encarregado de aprender sozinho como utilizá-las. Pressionando a tecla *Esc* outra vez, retorna uma janela, e observando, nota-se que o programa é salvo automaticamente para o disco rígido, sem necessidade de utilizar um comando para salvar. O mesmo acontece quando fecha-se o *menu* de configuração.

A.4 TRANSFERINDO O PROGRAMA E A CONFIGURAÇÃO

Até então foi programada a lógica a ser executada pelo CLP e configurado o *hardware*, com o *software* no modo de operação *OFFLINE*; para transferir o programa e a configuração ao CLP, o mesmo deve estar em modo de operação *ONLINE*, com o *software*.

Com o cabo de comunicação serial ligado entre o CLP e o *PC* pode-se estabelecer a comunicação; assim a transferência de dados para o CLP é possível, contudo é necessário primeiro colocar o *software* em comunicação com o CLP, e isto é feito pressionando a tecla *Alt+M*, até aparecer a mensagem de comunicação *ONLINE*. Se aparecer a mensagem *NO COMMUNICATIONS*, como mostrado na figura A.21, significa que ocorreu algum problema na comunicação, que pode ser na porta de comunicação *COM*, no cabo de

comunicação ou na opção de comunicação do CLP, quando esta não estiver corretamente configurada para comunicação serial.

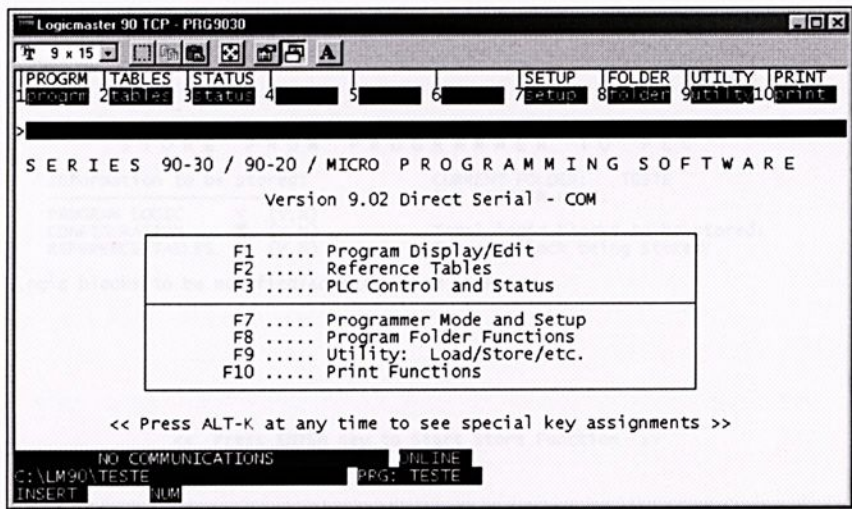


Figura A.21 - erro no modo de operação ONLINE.

Com o *micro* em comunicação *ONLINE* com o CLP, pode-se transferir o programa e a configuração para o CLP, através do *menu Utility (F9)*. Neste mesmo *menu* pode-se também, caso o CLP já tenha uma lógica armazenada em sua memória, transferir a lógica para o *PC*, através do comando *Load (F1)*. No entanto, a princípio o interesse é de enviar os dados referentes à programação e configuração, por isso, é necessário escolher o comando *Store*, que deverá armazenar (transferir) os dados do *PC* para o CLP.

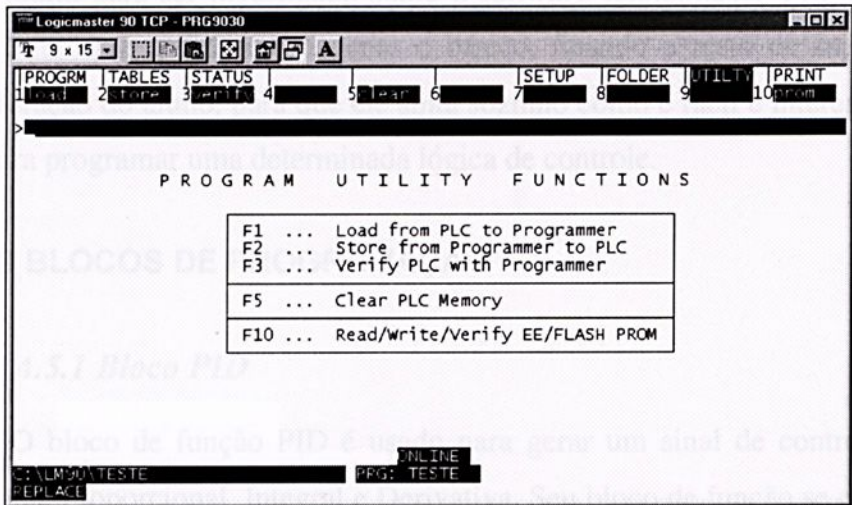


Figura A.22 - janela do utilitário de comunicação entre o software e o CLP.

Na próxima janela (figura A.23) seleciona-se os itens *PROGRAM LOGIC* e *CONFIGURATION* para serem transferidos ao CLP e logo após confirma-se, através da tecla



Enter. O programa perguntará se realmente deseja-se transferir o programa e configuração, e a resposta deve ser confirmativa.

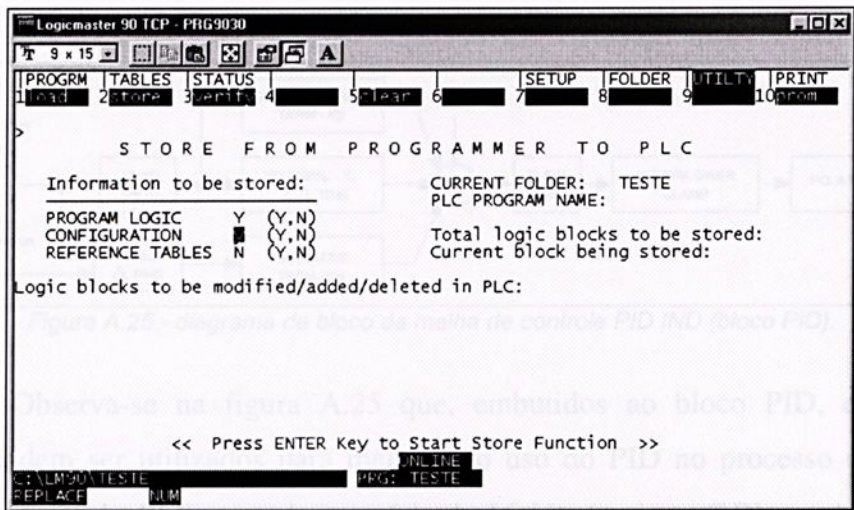


Figura A.23 - dados a serem transferidos para o CLP.

Quando a mensagem *Store complete* for mostrada, significa que o objetivo foi cumprido com êxito e aparecerá, no canto inferior direito da tela, a frase *LOGIC EQUAL*, informando que o programa armazenado na memória do CLP é o mesmo mostrado na tela do *software*.

Na pesquisa em laboratório foram verificadas quase todas ferramentas que o *software* possui e que, por conveniência, não foram explanadas aqui. Por isso, pode-se dizer que ainda há muito para ser investigado sobre o *software LogicMaster TCP/IP V9.02*, pois este trabalho limitou-se a descrever apenas o básico, ficando o resto de suas ferramentas, sujeitas a exploração do aluno, para que ele sinta sozinho como é fácil e interessante o uso de tal *software* para programar uma determinada lógica de controle.

A.5 ALGUNS BLOCOS DE PROGRAMAÇÃO

A.5.1 Bloco PID

O bloco de função PID é usado para gerar um sinal de controle, através da malha de controle Proporcional, Integral e Derivativa. Seu bloco de função se encontra dentro do *menu CONTRL* e pode-se utilizar a função *pidisa* (F4) ou o *pidind* (F5).

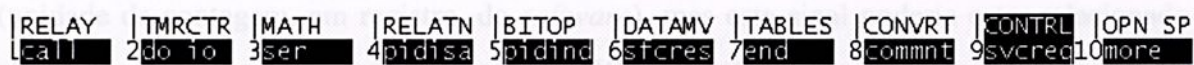


Figura A.24 - menu de instruções de controle.

O bloco PID deverá ser configurado, informando qual o sinal de referência (*SP*) para gerar o erro, qual a variável do processo (*PV*), que será comparada ao *SP*, e qual a variável de controle (*CV*), que será usada para realimentar a planta.

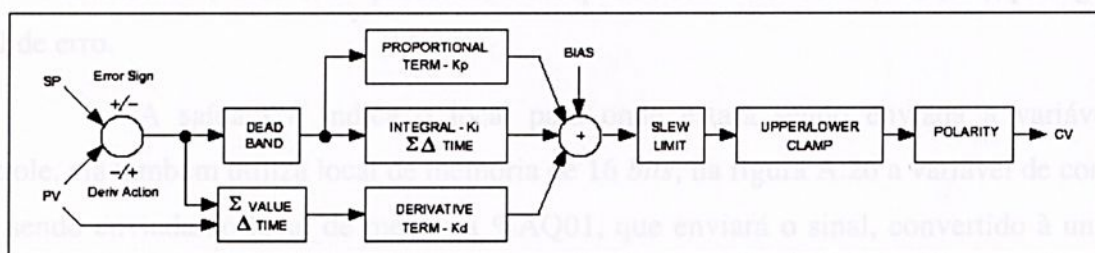


Figura A.25 - diagrama de bloco da malha de controle PID IND (bloco PID).

Observa-se na figura A.25 que, embutidos ao bloco PID, existem outros blocos, que podem ser utilizados para melhorar o uso do PID no processo em questão. O bloco PID necessita de 40 lugares de memória de 16 *bits* (registro %R) para armazenar toda sua configuração (ganhos proporcional, integral, derivativo, banda morta, grampos, etc).

Embora todos registros de sua configuração sejam importantes, serão explicados aqui apenas aqueles que necessitam de um enfoque maior (consulte o manual de referência *GFK-0467L* pág. 12-71 para maiores detalhes), mas antes de ajustar os registros do bloco PID, será feita uma pequena explanação sobre ele.

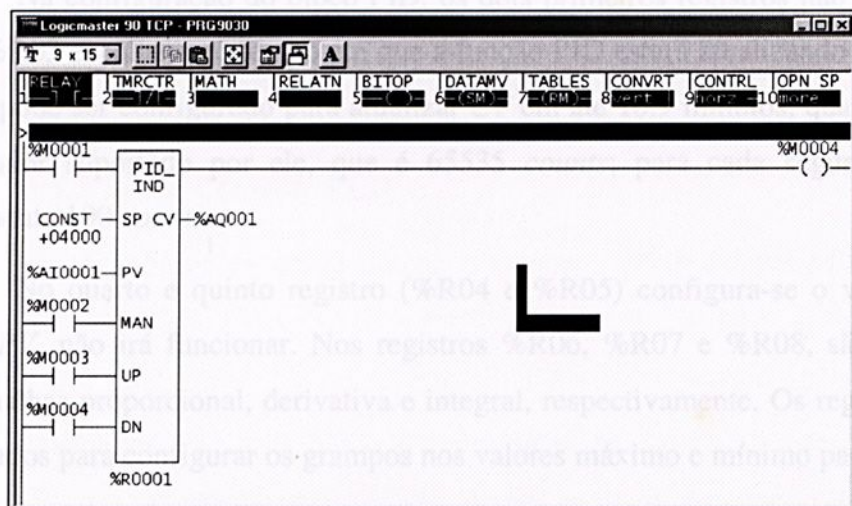


Figura A.26 - bloco PID IND.

No bloco PID, mostrado na figura A.26, o contato NA %M01 é usado para disponibilizar o uso da função PID. O sinal de referência (*SP*), deste bloco, é 4000 *counts* (unidade de contagem, em registro, do *software*), mas este sinal poderia estar relacionado a algum local de memória, por exemplo, %R100, ou até mesmo relacionado com um sinal

externo, através do módulo de entrada analógica e o local de memória referente a ela, por exemplo, %AI02. Na entrada *PV*, deste bloco, tem-se o local de memória %AI01 que, por exemplo, é um sinal gerado por um sensor ligado à planta; ele indica qual o valor atual da variável controlada (variável do processo) e compara o seu valor com 4000 (*SP*), para gerar o sinal de erro.

A saída *CV* indica o local para onde estará sendo enviada a variável de controle, ela também utiliza local de memória de 16 *bits*; na figura A.26 a variável de controle está sendo enviada ao local de memória %AQ01, que enviará o sinal, convertido à unidade utilizada pelo módulo, à saída analógica do CLP.

Embaixo do bloco de função escolhe-se qual o registro que servirá de referência para configuração do bloco PID; na figura A.26 foi utilizado o registro %R01. Por fim, têm-se as entradas *MAN*, *UP* e *DN*, que indicam respectivamente operação manual do PID, incremento e decremento do valor atual de *CV*, quando se utiliza a chave manual (*MAN*).

Como mostrado na figura A.26 o bloco de função PID tem sua referência no registro %R01, e será usada esta mesma referência de registro para explicar os principais tópicos da configuração do bloco PID. Logo os 40 registros que serão utilizados para configurar o PID são: %R01+0 até %R01+39, ou seja, %R01 até %R40.

Na configuração do bloco PID, os dois primeiros registros não são utilizados. No registro %R03 escolhe-se o tempo em que a função PID estará atualizando o valor de *CV*. Este registro pode ser configurado para atualizar *CV* em até 10.9 minutos, quando utilizamos o máximo valor suportado por ele, que é 65535 *counts*; para cada segundo ele utiliza aproximadamente 100 *counts*.

No quarto e quinto registro (%R04 e %R05) configura-se o valor da banda morta, onde *PV*, não irá funcionar. Nos registros %R06, %R07 e %R08, são inseridos os ganhos das malhas proporcional, derivativa e integral, respectivamente. Os registros %R10 e %R11 são usados para configurar os grampos nos valores máximo e mínimo para *CV*.

Esses são os principais registros configuráveis que o bloco PID utiliza para compor o sinal de controle, ficando os demais sujeitos a melhor interpretação estudando o manual do fabricante. O bloco PID só irá funcionar corretamente depois de ajustados os valores dos 40 registros que o configura. Sendo assim, é mostrado na figura A.27 como se pode *setar* os 40 registros, utilizando comandos da própria lógica para fazer isso.



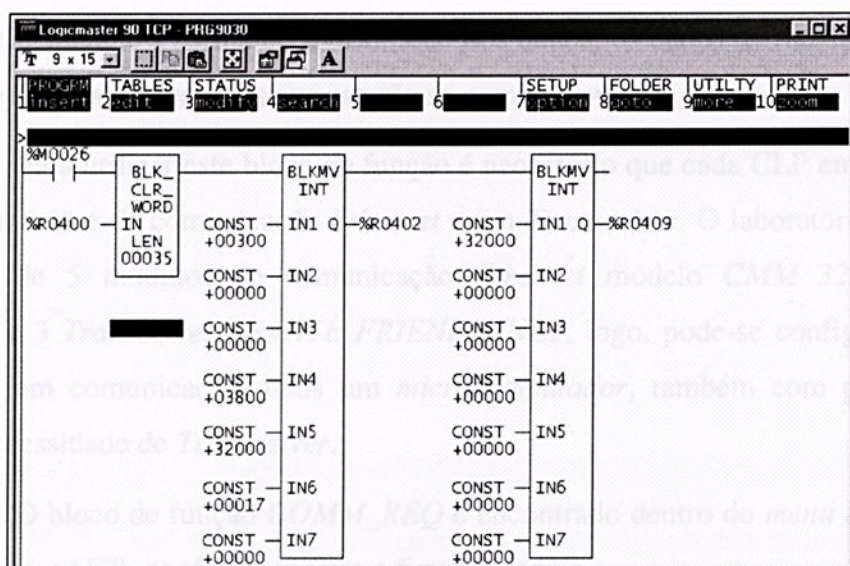


Figura A.27 - configuração dos registros do bloco PID IND.

Não é necessário que a chave %M26 esteja em nível lógico “1” durante a utilização do bloco PID, pois basta um pulso em %M26 para realizar a lógica mostrada acima, que transferirá dados de entrada para os locais de memória do tipo %R do CLP.

O bloco *BLK CLR_WORD* limpa uma quantidade de *LEN* palavras a partir do endereço informado na entrada *IN*; no exemplo mostrado na figura A.27, ele limpará os 35 registros logo após o registro %R400, incluindo o %R400. Depois de limpar os registros o bloco *BLK MV INT* move sete valores inteiros (*IN1* à *IN7*) para os próximos 7 locais de memória iniciando no registro da saída *Q* (%R402). Usa-se esta mesma metodologia para configurar os 40 registros necessários para o funcionamento do bloco PID.

A.5.2 Bloco *COMM_REQ*

Para este bloco sugere-se utilizar o manual *TCP/IP Ethernet Communications for the Series 90™-30 PLC GFK-1084B* a partir da página 3-6 para conhecer melhor o assunto.

Resumidamente, este bloco é responsável pela comunicação direta entre os dispositivos CLP, sem necessitar, por exemplo, de um *software* de supervisão que recebe a informação de um CLP e a envia para o outro. Assim como no bloco PID, este bloco também necessita de informações de determinados registros para configurar sua comunicação.

Basicamente o que é configurado através dos registros são: o tipo de comunicação (canal de escrita, de leitura, etc), o número do canal de comunicação (sendo que os CLPs envolvidos na comunicação não podem usar mais de um canal com o mesmo

número), a velocidade e a forma de transmissão dos dados, os dados de memória que estarão em comunicação e os dispositivos que estarão se comunicando.

Para utilizar este bloco de função é necessário que cada CLP em comunicação, possua uma interface de comunicação *Ethernet* e um *Transceiver*. O laboratório de ensino da FEIS dispõe de 5 módulos de comunicação *Ethernet* modelo *CMM 321 ETHERNET INTERFACE* e 3 *Transceiver ASANTÉ FRIENDLYNET*, logo, pode-se configurar uma rede com 3 CLPs em comunicação, mais um *microcomputador*, também com placa *Ethernet*, porém sem necessidade de *Transceiver*.

O bloco de função *COMM_REQ* é encontrado dentro do *menu DATAMV*, com o nome de *comreq (F7)*, conforme mostra a figura a seguir.



Figura A.28 - menu DATAMV (data move).

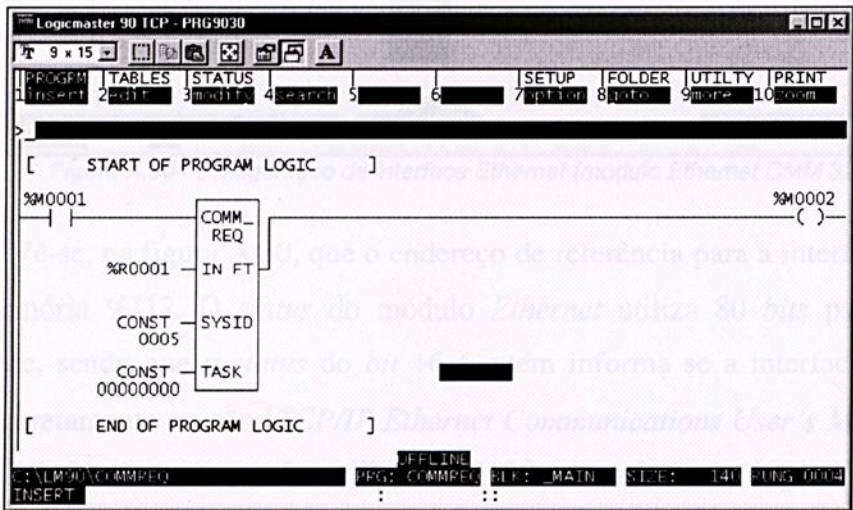


Figura A.29 - bloco de comunicação COMM_REQ.

Na figura A.29, mostrada acima, o bloco de comunicação entrará em funcionamento bastando um pulso da memória %M01. Caso aconteça algum problema com a comunicação, a memória %M02 será energizada (nível lógico “1”), podendo ser criado um intertravamento ou um sinal de alarme para tal situação.

Na entrada *IN* do bloco (figura A.29) insere-se o registro com o qual será iniciada a configuração do bloco de comunicação. Na entrada *SYSID* identifica-se em que *rack* e *slot* está localizada a interface de comunicação *Ethernet*. No caso apresentado na figura A.29, o módulo *Ethernet* está localizado na *rack* 00 e *slot* 05. A entrada *TASK* é sempre usada com valor 0 quando utiliza-se o CLP série 90-30.

Alguns detalhes são muito importantes para utilização da *COMM_REQ*, por exemplo, a *LAN* deverá estar funcionando para que a função *COMM_REQ* opere corretamente; assim pode-se colocar, como um dos intertravamentos para ligar a *COMM_REQ*, a condição da *LAN* estar funcionando corretamente. Quando configura-se o *hardware*, e conseqüentemente, a interface *Ethernet*, escolhe-se para ela um endereço de memória chamado *status* que informará se ela está funcionando corretamente ou não. Na figura A.30 é mostrado este endereço de *status*, ou endereço de referência.

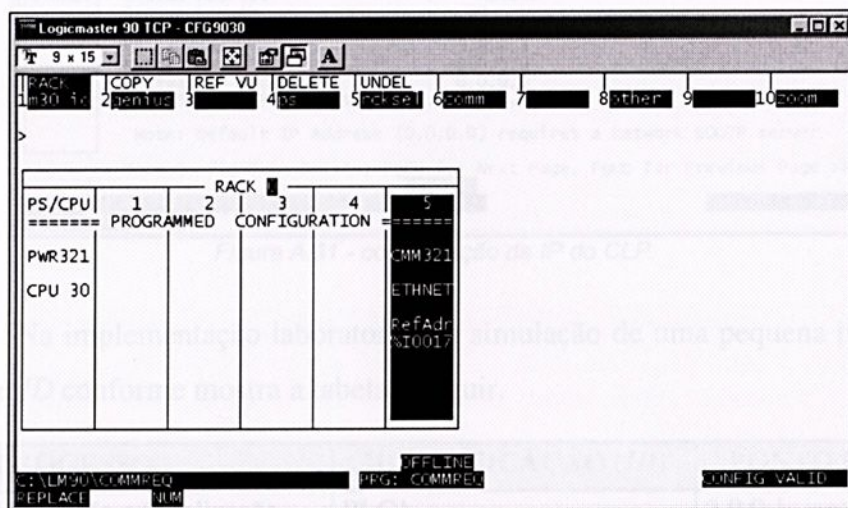


Figura A.30 - configuração da interface Ethernet (módulo Ethernet CMM 321).

Vê-se, na figura A.30, que o endereço de referência para a interface *Ethernet* é o local de memória %I17. O *status* do módulo *Ethernet* utiliza 80 bits para informar a situação da rede, sendo que o *status* do bit 16 é quem informa se a interface de rede está funcionando corretamente ou não [TCP/IP Ethernet Communications User's Manual – GFK-1084B pág. 3-34]. Sendo assim, se for utilizado %I17 como referência, logo %I32 é o bit que informa se a *LAN* está funcionando sem problemas ou não.

Desta forma, pode-se intertravar os blocos *BLKMOV INT* e *COMM_REQ* para funcionar somente se a rede estiver pronta para funcionar, bastando um pulso de leitura ou escrita, para iniciar a comunicação.

A.6 CONFIGURAÇÃO DOS CLPS

Cada CLP na rede recebe um *IP* e um nome de identificação *ID*. O número de *IP* de cada CLP é escolhido ao configurar as interfaces de comunicação *Ethernet*. Como exemplo, a figura A.31 mostra a configuração da interface *Ethernet* do controlador de nível,

cujo IP é 3.0.0.3. O nome de identificação ID é escolhido no menu CPU, porém o PC deverá estar em operação *ONLINE* para a escolha do nome.

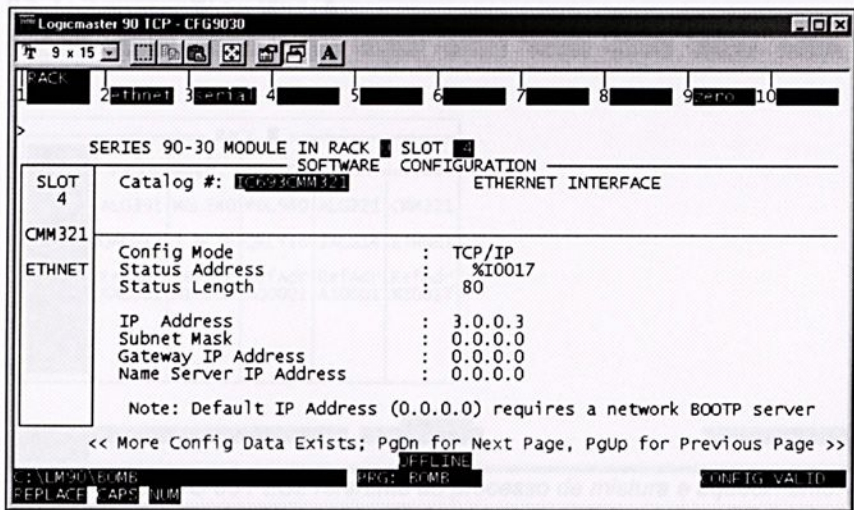


Figura A.31 - configuração da IP do CLP.

Na implementação laboratorial da simulação de uma pequena indústria, foram utilizados IP e ID conforme mostra a tabela a seguir.

PROCESSO	IDENTIFICAÇÃO (ID)	PONTO DE REDE (IP)
Sistema de Refrigeração e Sinalização	PLC1	3.0.0.1
Processo de Mistura e Aquecimento	PLC2	3.0.0.2
Controlador de Nível	PLC3	3.0.0.3
Micro Computador PC K6-2 450MHz	JEANRIBE	3.0.0.4

Tabela A.1 - IP e ID dos dispositivos ligados na rede LAN.

A seguir é mostrado, através de figuras, a configuração das I/O de todos CLPs que foram ligados em rede durante a implementação laboratorial.

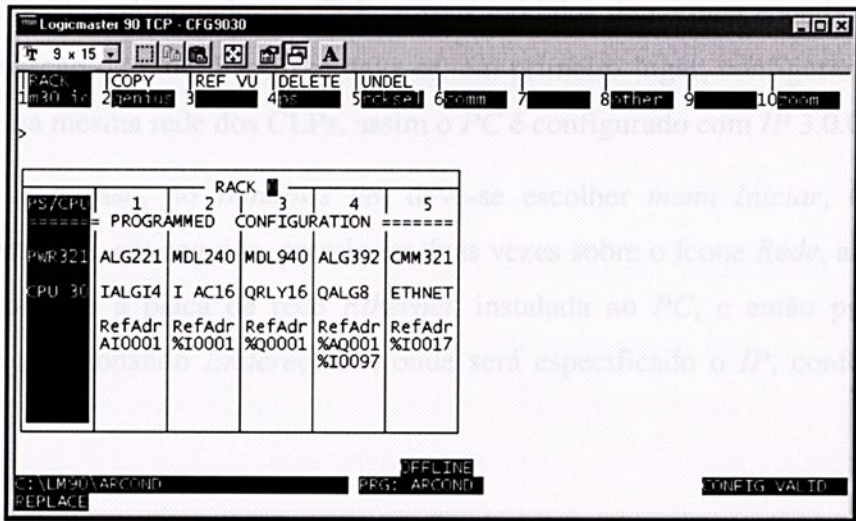


Figura A.32 - I/O do PLC1 referente ao sistema de refrigeração e sinalização via pêndulo invertido.

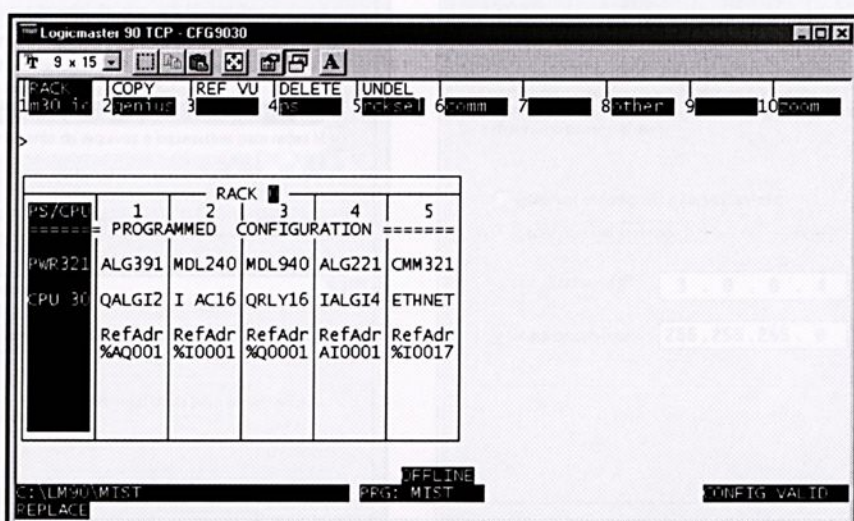


Figura A.33 – I/O do PLC2 referente ao processo de mistura e aquecimento.

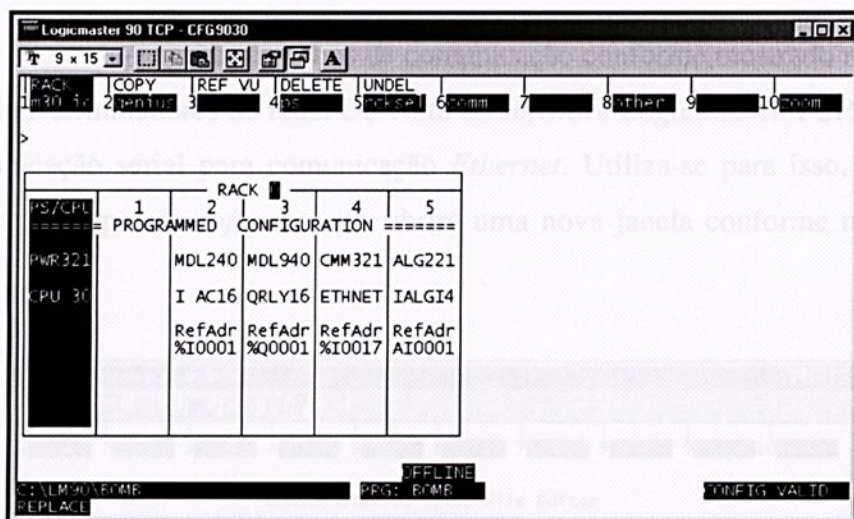
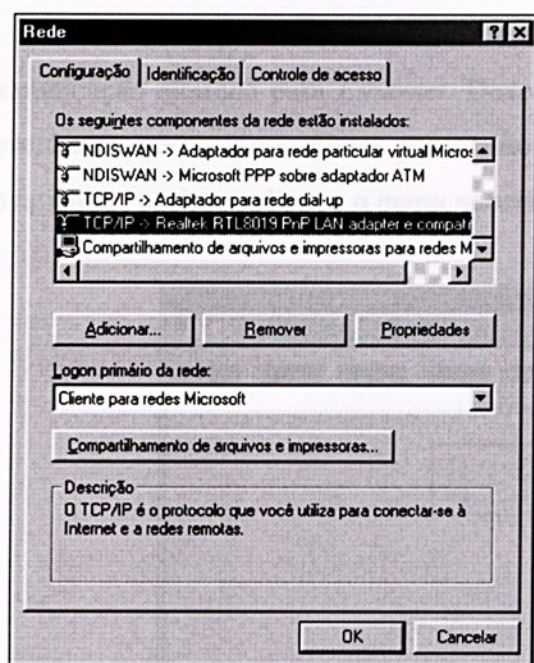


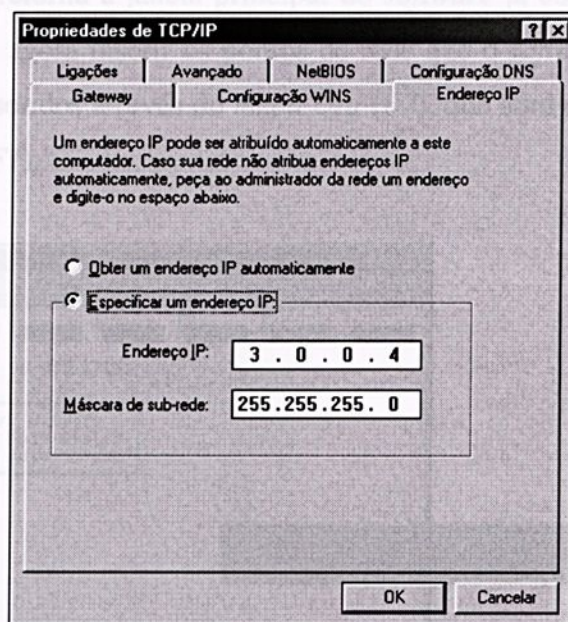
Figura A.34 - I/O do PLC3 referente ao controlador de nível.

Após configurados os CLPs e seus módulos de entrada e saída, parte-se então para a configuração da rede. Para isso deve-se, em primeiro lugar, configurar o computador para trabalhar na mesma rede dos CLPs, assim o PC é configurado com IP 3.0.0.4.

Para isso, no Windows 98, deve-se escolher menu *Iniciar*, *Configurações*, *Painel de Controle* e, em seguida, pressionar duas vezes sobre o ícone *Rede*, abrindo a janela de Rede. Escolhe-se a placa de rede *Ethernet*, instalada no PC, e então pressiona-se em *Propriedades*, selecionando *Endereço IP*, onde será especificado o IP, conforme mostra a figura A.35.



(a)



(b)

Figura A.35 - configuração do IP do computador. (a) janela de rede; (b) propriedades da placa de rede.

Conecta-se então os cabos de comunicação conforme mostrado na figura 5.13 e insere-se os dois terminadores de rede. De volta ao *software LogicMaster TCP/IP*, muda-se o tipo de comunicação serial para comunicação *Ethernet*. Utiliza-se para isso, o *menu Setup* (F9) da janela principal do *software*, que abrirá uma nova janela conforme mostra a figura A.36.

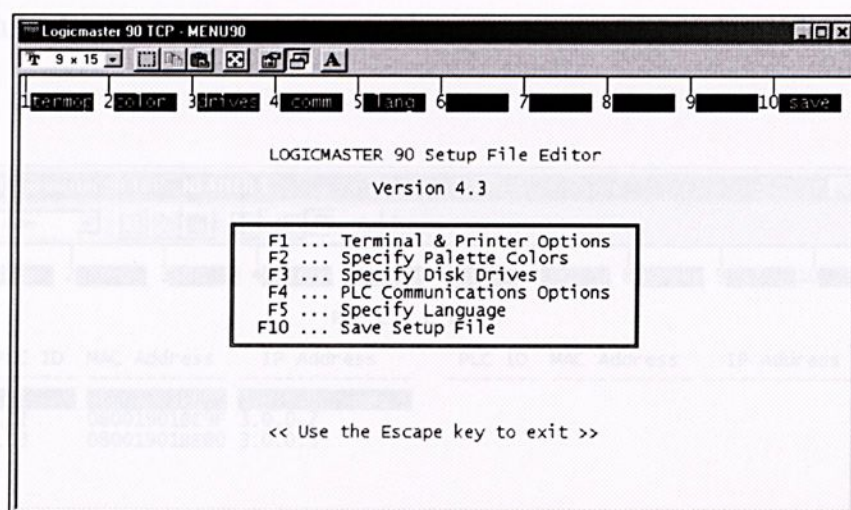


Figura A.36 - menu Setup para troca de comunicação serial para Ethernet.

Escolhe-se então o *menu PLC Communications Options* (F4), onde será mudado o tipo de comunicação serial para Ethernet. Pressionando *Esc*, volta-se para a janela mostrada na figura A.36, onde se deve salvar esta alteração, escolhendo a opção *save* (F10).

Ao pressionar *Esc* duas vezes retorna a janela principal do *software* já com a comunicação alterada para *Ethernet*. Deve-se agora inserir os pontos de rede que o *software* deverá estar gerenciando. Esses pontos são inseridos através do *menu Útil (F7)*, que abre uma nova janela, onde escolhe-se o *menu netwrk (F1)*.

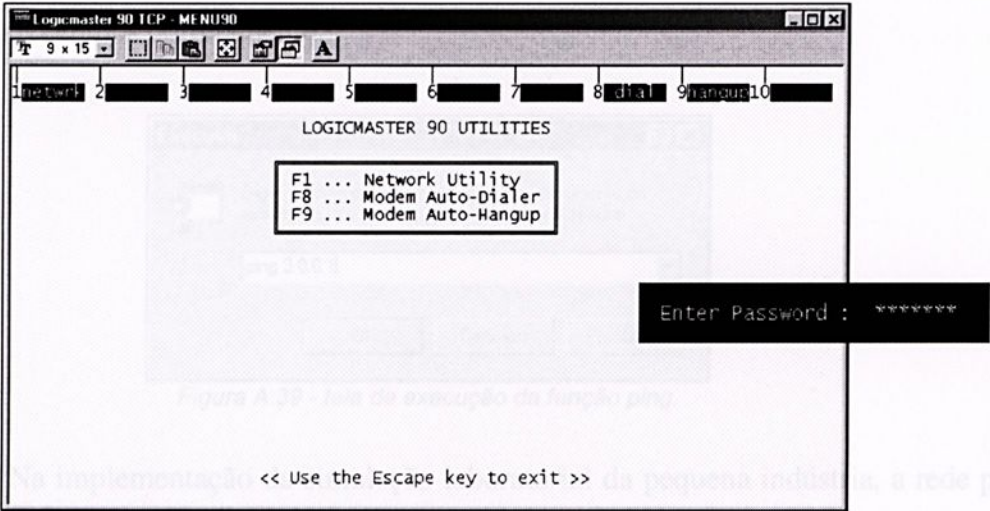


Figura A.37 - janela de configuração da rede de CLPs.

O *software* pedirá para entrar com a senha e deve-se digitar “netutil” (sem as aspas). Pressionando a tecla *Enter* abrirá a nova janela onde se seleciona *PLC List (F1)*. Adiciona-se os CLPs que estarão em rede (no caso da implementação da simulação industrial apenas 3 CLPs), como mostra a figura A.38. A opção *MAC Address* de cada CLP está gravada junto ao módulo de comunicação *Ethernet* (deve-se retirar a tampa do módulo para ler o *Mac Address*).

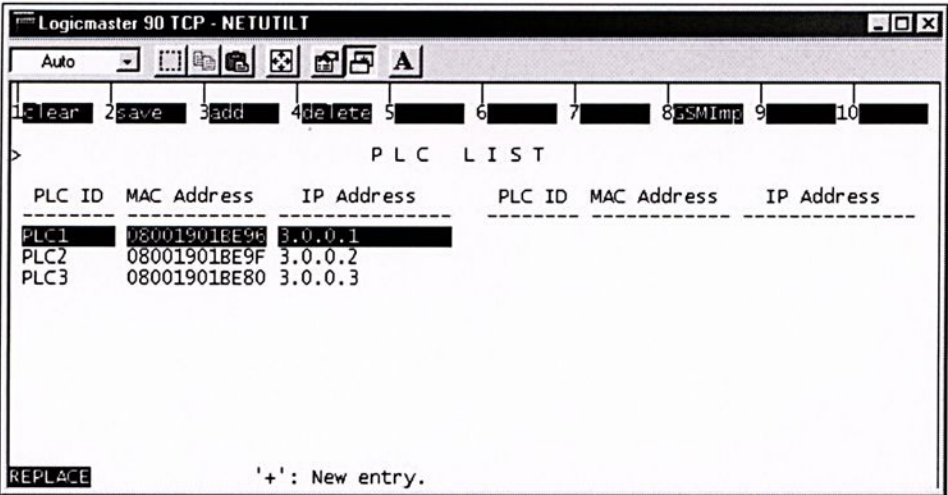


Figura A.38 - dispositivos (CLPs) ligados a rede LAN.

Após inserir os CLPs, deve-se salvar esta configuração escolhendo a opção *save* (F2). Pressiona-se *Esc* até voltar ao *menu* principal, já em comunicação *Ethernet*. Para testar se a rede está funcionando corretamente podemos usar o comando *ping*, a partir do computador. No *desktop* do *Windows* escolha *menu* *Iniciar*, *Executar* e então digita-se *ping 3.0.0.1* (figura A.39), depois *ping 3.0.0.2* e por último *ping 3.0.0.3*, verificando se os dados são enviados e recebidos corretamente.

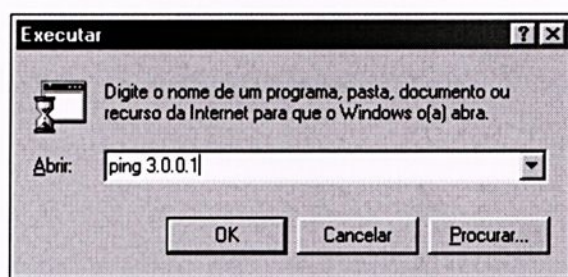


Figura A.39 - tela de execução da função *ping*.

Na implementação da simulação laboratorial da pequena indústria, a rede para comunicação entre os 3 CLPs e o computador, foi a *Ethernet*, que utiliza o protocolo de comunicação *TCP/IP*. Ela foi implementada na topologia *bus*, utilizando cabo coaxial para interligação entre os pontos da rede. Para receber e enviar os sinais para a rede, os CLPs utilizam o dispositivo chamado *transceiver*, já o *microcomputador*, também ligado à rede, não necessita de *transceiver*, pois a própria placa de rede *Ethernet* se encarrega de condicionar os sinais para envio e recebimento de dados.

DETALHES DOS PROCESSOS UTILIZADOS NA SIMULAÇÃO LABORATORIAL DA INDÚSTRIA

B.1 PROCESSO DE MISTURA E AQUECIMENTO DE LÍQUIDOS

O processo de mistura de líquido é o principal processo entre todos segmentos desta pequena indústria. Ele é responsável direto pelo produto final do todo sistema automatizado, por isso é importante conhecer tal processo individualmente antes de inserir os intertravamentos com os demais processos controlados, e a etapa de aquecimento da mistura.

Apêndice B

Detalhes dos processos utilizados na simulação laboratorial da indústria

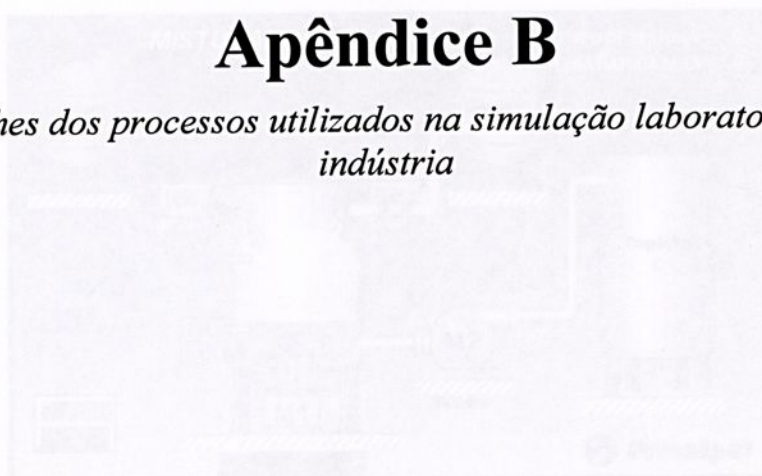


Figura B.1 - janela de supervisão e controle do processo de mistura em operação.

A princípio este processo contará com dois motores de indução trifásicos: M1 e M2. O motor M1 é responsável pela mistura dos líquidos e o motor M2 funciona como uma bomba que transfere o líquido de um reservatório para outro.

O motor M1 será ligado com carga, logo sua partida (início de funcionamento) ocorre com as bobinas configuradas em estrela (tensão reduzida) e passará para triângulo (tensão plena) para operação em regime, evitando assim que a corrente de partida seja muito alta. As bobinas também podem ser configuradas de forma a inverter o sentido de rotação do eixo do rotor, bastando para isso a troca de duas fases.

DETALHES DOS PROCESSOS UTILIZADOS NA SIMULAÇÃO LABORATORIAL DA INDÚSTRIA

B.1 PROCESSO DE MISTURA E AQUECIMENTO DE LÍQUIDOS

O processo de mistura de líquido é o principal processo entre todos segmentos desta pequena indústria. Ele é responsável direto pelo produto final de todo sistema automatizado, por isso é importante conhecer tal processo individualmente antes de inserir os intertravamentos com os demais processos controlados, e a etapa de aquecimento da mistura.

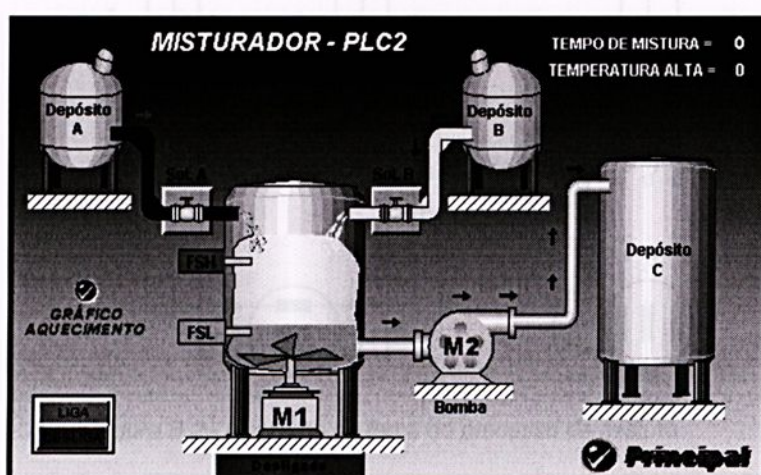


Figura B.1 - janela de supervisão e controle do processo de mistura em operação.

A princípio este processo contará com dois motores de indução trifásicos: M1 e M2. O motor M1 é responsável pela mistura dos líquidos e o motor M2 funciona como uma bomba que transfere o líquido de um reservatório para outro.

O motor M1 será ligado com carga, logo sua partida (início de funcionamento) ocorre com as bobinas configuradas em estrela (tensão reduzida) e passará para triângulo (tensão plena) para operação em regime, evitando assim que a corrente de partida seja muito alta. As bobinas também podem ser configuradas de forma a inverter o sentido de rotação do eixo do rotor, bastando para isso a troca de duas fases.

A partida do motor M2 ocorre de forma direta na configuração estrela, visto que a carga a ser vencida, é muito baixa, pois a própria pressão, exercida pela força peso do líquido misturado, ajuda na transferência do líquido. Como não há necessidade de inversão do sentido de rotação do eixo deste motor, nem mudança na configuração das bobinas, mostra-se que o circuito elétrico de força para ligação deste motor é bastante simples, bastando apenas um contator (figura B.2).

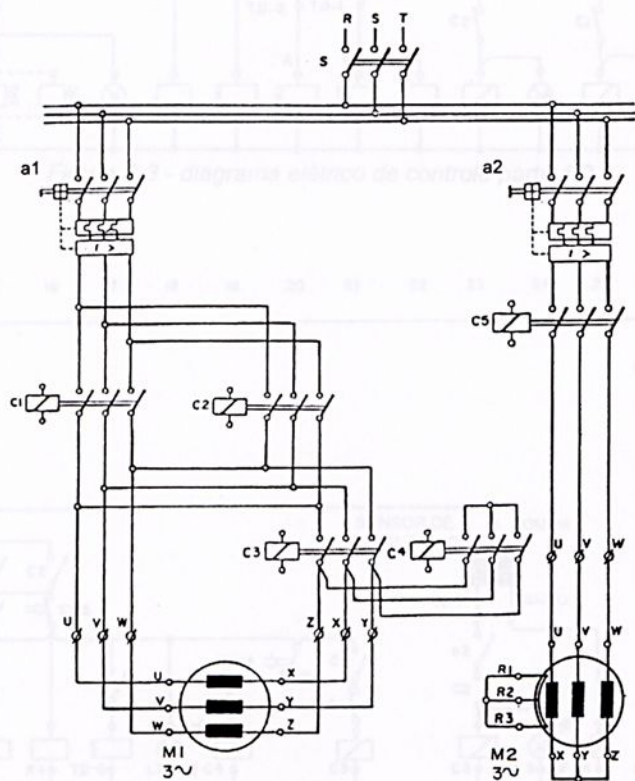


Figura B.2 - diagrama de força do processo de mistura.

A figura B.2 mostra o circuito de força, com a ligação dos contatores, e os relés de sobrecarga, inseridos antes dos contatores, para evitar danos devido a sobrecarga de corrente nos enrolamentos dos motores.

B1.1 Lógica de Controle Utilizando Diagrama Elétrico

Descreve-se a seguir o processo automático de controle do misturador de líquidos, através da análise do diagrama elétrico de controle, representado nas figuras B.3 e B.4.

Em primeiro lugar, é importante salientar que, no diagrama elétrico, todos contatos a1 e a2 estão representados desligados, ou seja, antes de iniciarmos o processo de mistura será necessário ligar o relé de sobrecarga e, conseqüentemente, mudar o estado (aberto ou fechado) dos contatos a1 e a2. Assim, os contatos que estão representados fechados se abrem e os contatos que estão representados abertos se fecham. Estes componentes atuam em caso de sobrecarga de corrente no motor, durante a etapa de mistura ou na etapa de bombeamento dos líquidos

Porém, na prática, os relés a1 e a2 (figura B.2) disponíveis, no laboratório de ensino da FEIS, não possuem contatos auxiliares, logo, todos contatos referentes a estes relés, no diagrama elétrico das figuras B.3 e B.4, serão retirados durante a implementação laboratorial.

Na primeira coluna, do diagrama elétrico de comando (figura B.3), tem-se duas botoeiras: PB-1 e PB-2. Pressionando a botoeira PB-2 dá-se início ao processo, que energizará a bobina do relé R1; logo o contato NA de R1 se fecha, selando a botoeira PB-2. Estando o sensor de nível alto na posição BAIXO, as válvulas solenóides são energizadas, abrindo-se e permitindo o enchimento do reservatório. Estas válvulas solenóides são representadas por duas lâmpadas num painel sinóptico (S_A e S_B), que estarão acesas durante todo processo de enchimento.

Encerrado o processo de enchimento, ou seja, supondo o reservatório cheio, o sensor de nível alto passa da posição BAIXO para ALTO, logo as válvulas solenóides são desenergizadas, interrompendo o fluxo de líquido dos depósitos A e B para o reservatório. Neste momento inicia-se o processo de mistura, que energizará a bobina do relé R2, fechando o contato NA de R2 (coluna 5), mantendo um selo sobre o sensor de nível alto. Ao mesmo instante, é energizado a bobina do relé de retardo na energização TD-3, que está ajustado para atuar após 40 segundos, finalizando o processo de mistura.

Enquanto TD-3 não atua, o processo de mistura entre os dois líquidos não pára, logo o relé de tempo TD-1 (tempo em que o motor estará girando num sentido) será energizado. Como TD-1 é um relé de retardo na energização, este só atuará após transcorrido seu tempo de ajuste (12 segundos); enquanto isto não ocorre, as bobinas de TD-2 e R3 se manterão desenergizados devido ao contato aberto de TD-1 (coluna 8).

Logo, a bobina do contator C1 será energizada e a lâmpada L5 acenderá, representando que o eixo do motor gira no sentido horário¹. Estando o contator C1 energizado, pode-se verificar, da análise do diagrama do circuito de força (figura B.2), que o motor M1 está pronto para ser ligado em Y ou Δ dependendo apenas dos contadores C4 e C3. A lâmpada L2, na coluna 14, estará acesa durante toda etapa de mistura.

Continuando a análise, supõe-se que TD-1 ainda não atuou, então observa-se na coluna 15 (figura B.4), que o contato NA de C1 estará fechado devido a energização da bobina de C1 (coluna 10). Assim, será energizado o relé de tempo TD-4 que permitirá a ligação do motor (M1) em Y durante um tempo pré-fixado.

Na coluna 19 observa-se que a bobina do contator C4 é energizada, logo, analisando o circuito de força, verifica-se que o motor M1 parte na configuração Y. A bobina do relé R4 (coluna 16) também estará energizada, fechando os contatos NA de R4, que selará o contato NF de TD-5 e energizará a bobina do relé de retardo na desenergização TD-5. Logo, estando a bobina de TD-5 energizada, o contato NF de TD-5, que foi selado pelo contato de R4 (coluna 15), se abre imediatamente. A lâmpada L7 estará acesa durante todo o tempo em que o motor M1 estiver ligado, independente da configuração estrela ou triângulo.

Tendo-se passado o tempo de ajuste do relé TD-4 (5 segundos), seu contato auxiliar atua, desenergizando a bobina do contator C4 e energizando a bobina do contator C3, mudando assim, a configuração das bobinas do motor M1 para a ligação em Δ (tensão plena). É importante notar que, ao ser energizada a bobina de C3, a bobina de TD-4 é desenergizada (ver coluna 15), logo o contato de TD-4, da coluna 20, volta para a posição inicial, mas a bobina de C3 continua energizada através do caminho representado na coluna 21.

Para garantir o intertravamento de C3 e C4 é necessário colocar em série com a bobina de C4 um contato NF de C3 e em série com a bobina de C3 um contato NF de C4, evitando assim a energização mútua das duas bobinas, que ocasionaria um curto entre as 3 fases.

Até o presente, foi analisada a partida da máquina em Y- Δ , tendo-se então que analisar o momento em que ocorre a mudança no sentido de rotação do eixo do motor M1, ou seja, a troca de duas fases na alimentação do motor. O tempo necessário para esta troca é dado pelo relé temporizador TD-1 (12 segundos depois de iniciado a etapa de mistura).

¹ É adotado sentido horário durante a análise, mas este sentido é determinado dependendo da ligação do circuito de força.



Transcorrido este tempo, o contato NA de TD-1 na coluna 8 se fecha energizando a bobina de TD-2, que iniciará contagem de tempo, também de 12 segundos. Ao mesmo tempo, a bobina do relé R3 é energizada, mudando-se a posição do contato R3 (coluna 11), energizando assim a bobina do contator C2.

Pode-se verificar, no diagrama de força da figura B.2, que quando C2 é energizado e C1 desenergizado, inverte-se a posição de duas fases, ocasionando a inversão do sentido de giro do eixo do motor.

A inversão no sentido de giro do motor M1 não pode ocorrer enquanto o mesmo gira com velocidade plena num sentido, pois a carga a ser vencida seria muito grande, acarretando num aumento considerável da corrente durante a inversão, que não é interessante. Portanto, para inverter o sentido de rotação do eixo do motor, deve-se primeiro desligá-lo e aguardar que pare, para então religá-lo numa configuração que garanta a inversão do sentido de rotação.

Para desligar o motor usa-se um pequeno detalhe que às vezes passa despercebido, devido à velocidade de sua ocorrência: o tempo de chaveamento de um dispositivo eletromecânico. Durante o processo transitório de chaveamento do contato de R3 (coluna 11), tem-se um pequeno intervalo de tempo, chamado de tempo de chaveamento, em que este contato se mantém aberto, desenergizando tanto a bobina do contator C1 como a do contator C2. Este pequeno intervalo de tempo é suficiente para manter aberto os contatos NA de C1 e C2 nas colunas 15 e 16 respectivamente, desenergizando as bobinas de R4 e TD-5.

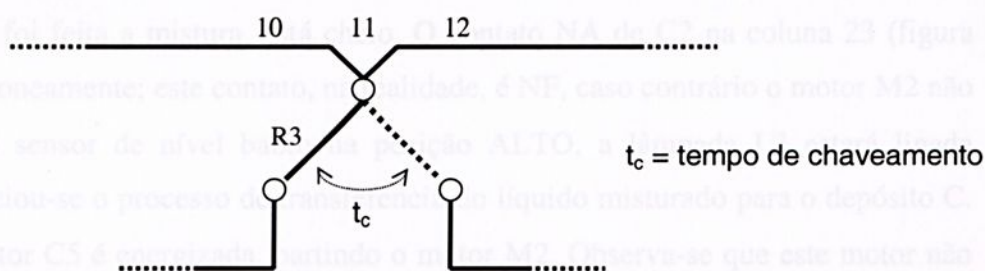


Figura B.5 - chaveamento do contato de R3 da posição fechada para aberto.

O contato NF de TD-5 (coluna 16) estava aberto devido a energização da bobina de TD-5 quando da análise de rotação no sentido horário, porém agora, após a desenergização da bobina de TD-5, este contato não volta imediatamente à condição inicial NF, pois se trata de um relé de retardo na desenergização. Logo ele aguardará seu tempo de ajuste (5 segundos) para depois se fechar novamente.

Com isso, o motor estará momentaneamente desligado, pois mesmo depois que o contato NA de C2 (ou C1) na coluna 15 (coluna 16) se fechar, não tem-se caminho para energizar o contator C3, visto que a bobina de R4 está desenergizada e o contato NF de TD-5 só se fechará novamente depois de 5 segundos (tempo necessário para o motor parar). O estado de funcionamento do motor M1 é descrito pela lâmpada L7.

Passado os 5 segundos da desenergização da bobina de TD-5 seu contato NF novamente se fecha, acendendo novamente a lâmpada L7 e iniciando novamente o processo de partida do motor M1 em Y, mas agora com o eixo do motor girando no sentido anti-horário, repetindo-se então toda análise feita quando o eixo do motor girava no sentido horário.

Decorrido o tempo de ajuste do dispositivo TD-2, a bobina de TD-1 é desenergizada (coluna 7), voltando seu contato NA da coluna 8 para a posição aberto, desenergizando as bobinas de R3 e TD-2. Assim, novamente energiza-se a bobina de TD-1 (coluna 7), iniciando a nova etapa de partida do motor em Y- Δ , novamente no sentido horário.

A etapa de mistura ocorre durante todo o tempo em que deseja-se que os líquidos A e B sejam misturados. Quando TD-3 atuar (40 segundos após o início da mistura) seu contato NF, da coluna 7, será chaveado para posição NA, iniciando então o processo de atuação da bomba (motor M2), que transferirá todo o líquido misturado para um novo reservatório.

Inicialmente o sensor de nível baixo estará acusando nível ALTO, já que o reservatório, onde foi feita a mistura, está cheio. O contato NA de C2 na coluna 23 (figura B.4), é descrito erroneamente; este contato, na realidade, é NF, caso contrário o motor M2 não partirá. Estando o sensor de nível baixo na posição ALTO, a lâmpada L3 estará ligada mostrando que iniciou-se o processo de transferência do líquido misturado para o depósito C. A bobina do contator C5 é energizada, partindo o motor M2. Observa-se que este motor não precisa de partida Y- Δ , já que a carga mecânica é baixa, pois a própria pressão do líquido força sua transferência para o depósito C.

Após transferido todo o líquido de um reservatório para o outro, o sensor de nível baixo passa da posição ALTO para BAIXO, desligando a lâmpada L3, desenergizando a bobina do contator C5 e, conseqüentemente, desligando o motor M2. A lâmpada L4 indica o final de todo o processo de mistura.

Pode-se parar todo o processo bastando pressionar a botoeira PB-1, não importando qual etapa do processo esteja em funcionamento (enchimento, mistura ou transferência dos líquidos).

B.1.2 Lógica de Controle Utilizando Diagrama Ladder

Foi descrito o processo de controle utilizando dispositivos eletromecânicos e não inserindo os intertravamentos que o processo de mistura faz com os outros dois processos: Controle de Nível dos depósitos A e B e o Sistema de Refrigeração do motor M1, respectivamente controlados pelos CLPs 1 e 3.

No diagrama *Ladder*, apresentado nas figura B.6 à B.9 e controlado pelo CLP2, é inserido esses intertravamentos, bem como acrescenta-se ao processo de mistura uma nova etapa: aquecimento do líquido misturado. Ou seja, depois de misturados os líquidos A e B, eles serão aquecidos e só depois serão transferidos para o depósito C.

Para efetuar o aquecimento, o CLP2 produzirá, através de um controle Proporcional, Integral e Derivativo (PID), uma variável de controle com valor máximo de 5Volts, que passará por um amplificador de potência e alimentará uma resistência contida num tanque com água (reservatório de mistura). Insere-se neste tanque um sensor de temperatura que informará, através de um sinal analógico, ao CLP2 o valor atual da temperatura (variável do processo) para que o PID funcione corretamente. A variável de controle é armazenada na memória %AQ01 e a variável do processo é armazenada na memória %AI01 do CLP2.

Figura B.6 - diagrama Ladder do processo de mistura e aquecimento - parte 1/4

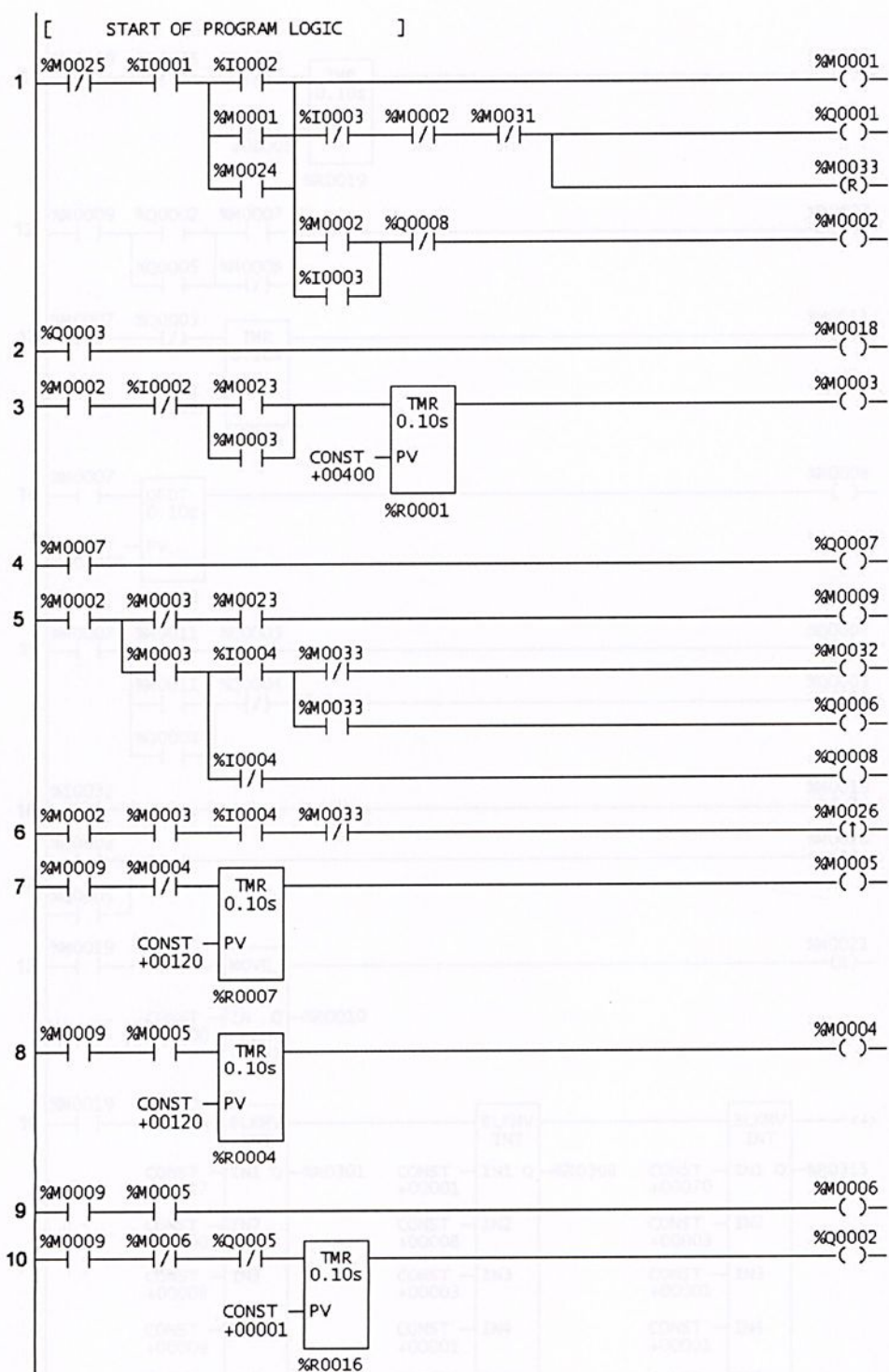


Figura B.6 - diagrama Ladder do processo de mistura e aquecimento— parte 1/4.

Figura B.7 - diagrama Ladder do processo de mistura e aquecimento— parte 2/4.

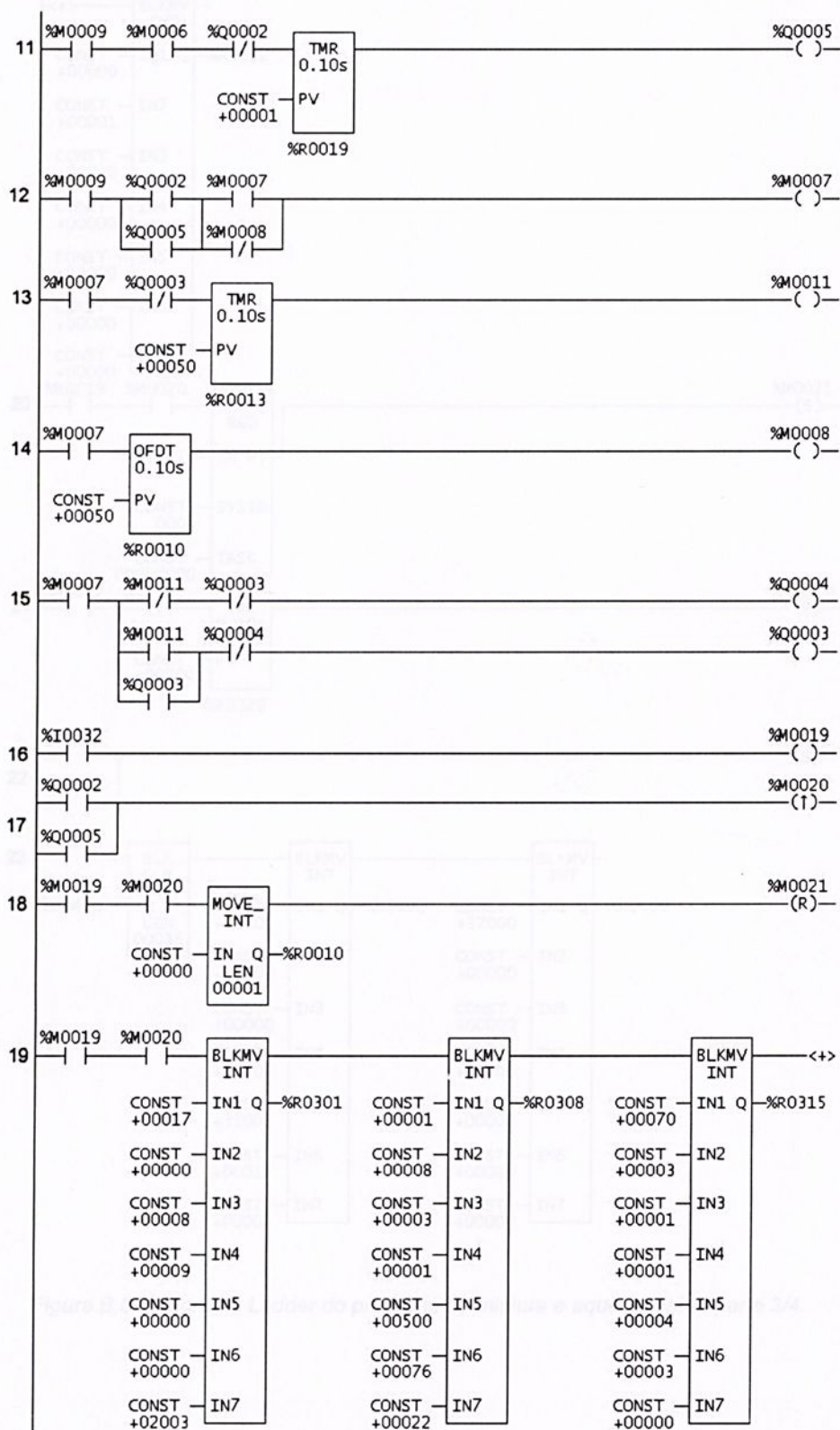


Figura B.7 - diagrama Ladder do processo de mistura e aquecimento- parte 2/4.

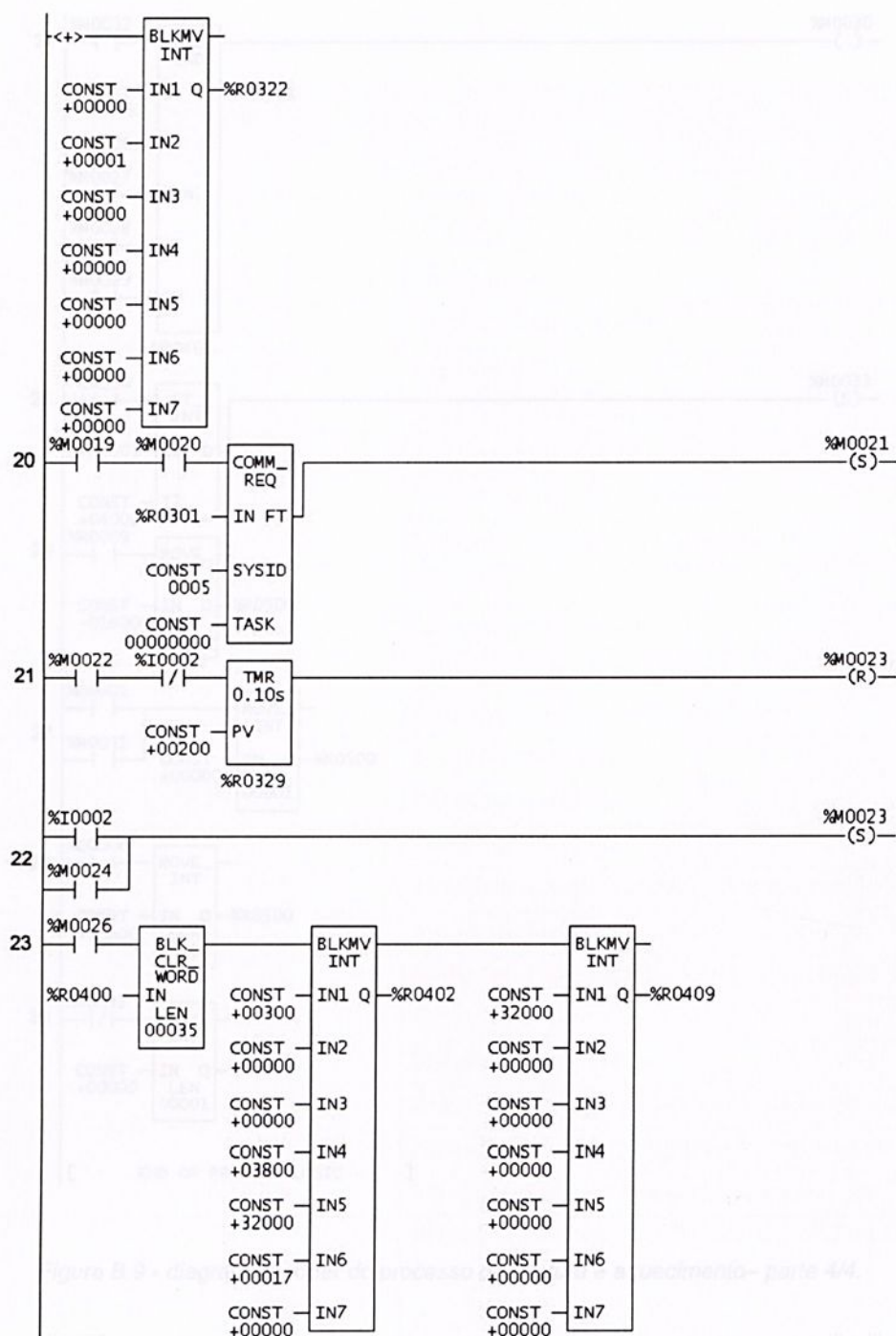


Figura B.8 - diagrama Ladder do processo de mistura e aquecimento— parte 3/4.

ENDEREÇO	CONTEÚDO	FUNÇÃO
%M001	Botão PB-1	Desligar o processo de mistura
%M002	Botão PB-2	Ligar o processo de mistura
%M003	Sensor de nível alto	Sensor de nível alto

Tabela B.1 - dados de memória utilizados na lógica de programação - parte 1/3.

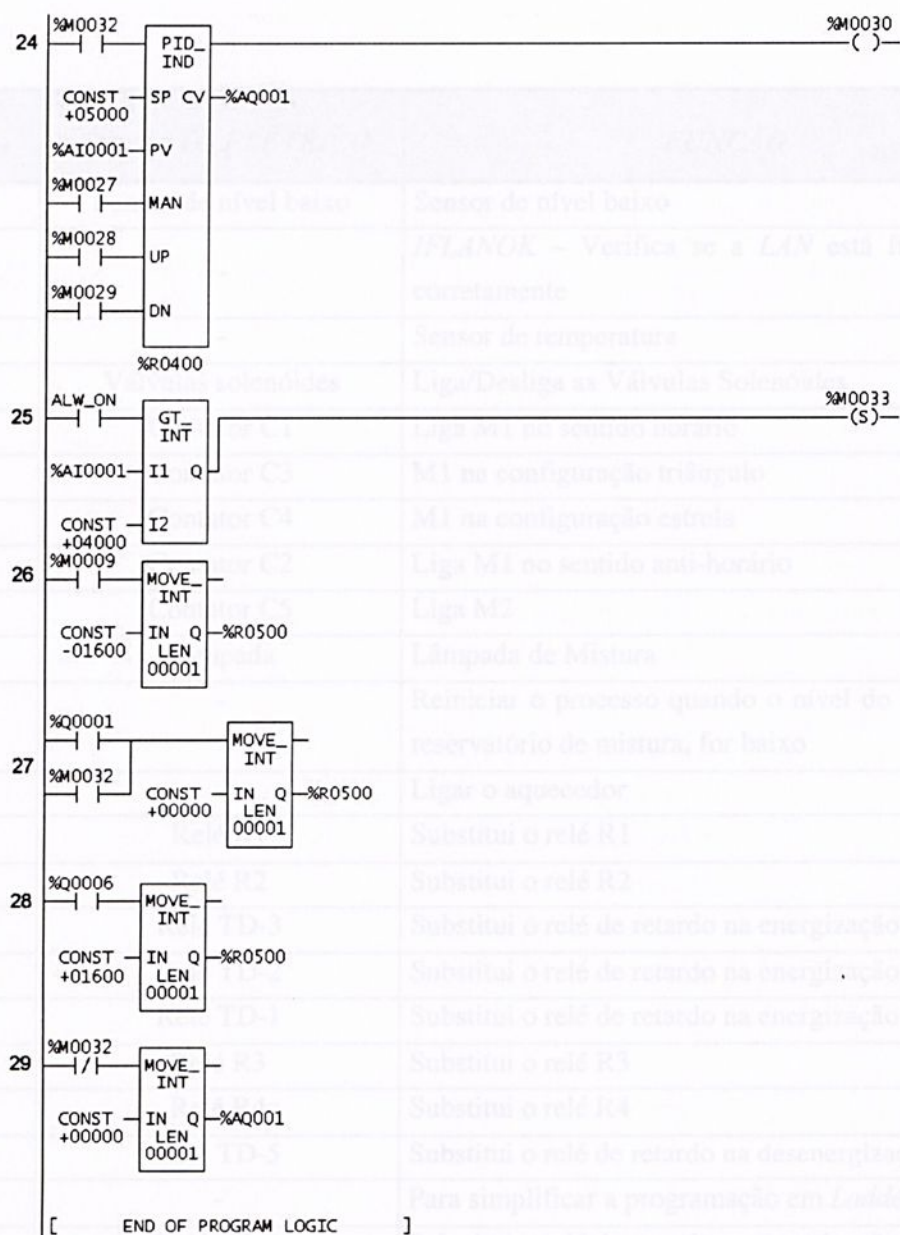


Figura B.9 - diagrama Ladder do processo de mistura e aquecimento— parte 4/4.

As tabela B.1 à B.3, apresentam todos locais de memória, utilizados na lógica de programação *Ladder*, seu contato elétrico referente (se existir) e a sua função

LOCAL DE MEMÓRIA	CONTATO ELÉTRICO	FUNÇÃO
%I01	Botoeira PB-1	Desligar o processo de mistura
%I02	Botoeira PB-2	Ligar o processo de mistura
%I03	Sensor de nível alto	Sensor de nível alto

Tabela B.1 - locais de memória utilizados na lógica de programação – parte 1/3.

LOCAL DE MEMÓRIA	CONTATO ELÉTRICO	FUNÇÃO
%I04	Sensor de nível baixo	Sensor de nível baixo
%I32	-	IFLANOK – Verifica se a LAN está funcionando corretamente
%AI01	-	Sensor de temperatura
%Q01	Válvulas solenóides	Liga/Desliga as Válvulas Solenóides
%Q02	Contator C1	Liga M1 no sentido horário
%Q03	Contator C3	M1 na configuração triângulo
%Q04	Contator C4	M1 na configuração estrela
%Q05	Contator C2	Liga M1 no sentido anti-horário
%Q06	Contator C5	Liga M2
%Q07	Lâmpada	Lâmpada de Mistura
%Q08	-	Reiniciar o processo quando o nível do líquido, no reservatório de mistura, for baixo
%AQ01	-	Ligar o aquecedor
%M01	Relé R1	Substitui o relé R1
%M02	Relé R2	Substitui o relé R2
%M03	Relé TD-3	Substitui o relé de retardo na energização TD-3
%M04	Relé TD-2	Substitui o relé de retardo na energização TD-2
%M05	Relé TD-1	Substitui o relé de retardo na energização TD-1
%M06	Relé R3	Substitui o relé R3
%M07	Relé R4	Substitui o relé R4
%M08	Relé TD-5	Substitui o relé de retardo na desenergização TD-5
%M09	-	Para simplificar a programação em <i>Ladder</i>
%M11	Relé TD-4	Substitui o relé de retardo na energização TD-4
%M18	-	Indica se a bobina do contator C3 está energizada
%M19	-	Energiza se a LAN estiver ok
%M20	-	Para ligar a <i>COMM_REQ</i>
%M21	-	Indica falta na <i>COMM_REQ</i>
%M22	-	Verifica o estado do termostato T2 (CLP1)
%M23	-	Desliga a mistura se a temperatura do termostato T2 estiver alta por mais de 20 segundos
%M24	-	Liga o processo de mistura através do <i>software Cimplicity HMI</i>

Tabela B.2 - locais de memória utilizados na lógica de programação – parte 2/3.



LOCAL DE MEMÓRIA	CONTATO ELÉTRICO	FUNÇÃO
%M25	-	Desliga o processo de mistura através do <i>Cimplicity HMI</i>
%M26	-	Ajustar as configurações do PID
%M27	-	Liga a chave manual do PID
%M28	-	Ajusta o PID em <i>up</i>
%M29	-	Ajusta o PID em <i>down</i>
%M30	-	Informa se o PID está ok
%M31	-	Intertravamento com o CLP3
%M32	-	Liga o sistema de aquecimento
%M33	-	É energizado quando a temperatura da mistura atinge o valor desejado
%R01	-	Registro de temporizador
%R04	-	Registro de temporizador
%R07	-	Registro de temporizador
%R10	-	Registro de temporizador
%R13	-	Registro de temporizador
%R16	-	Registro de temporizador
%R19	-	Registro de temporizador
%R301- %R323	-	Configura a <i>COMM_REQ</i>
%R329	-	Registro de temporizador
%R400- %R415	-	Configura o bloco PID
%R500	-	Registra a etapa do processo de mistura

Tabela B.3 - locais de memória utilizados na lógica de programação – parte 3/3

Comparando o diagrama *Ladder* das figuras B.6 à B.9, referente ao controle do processo de mistura e aquecimento, com o diagrama elétrico de controle (figuras B.3 e B.4), pode-se facilmente identificar muitos pontos escritos em *Ladder*, devido a semelhança das duas formas de escrever a lógica de controle. Por exemplo, a botoeira normalmente fechada PB-1 é a descrita no *Ladder* através do contato NA¹ %I01, outro exemplo é o *bit* %M01, que está relacionado ao relé R1.

Logo, pode-se facilmente identificar os contatos, chaves, sensores e as bobinas, através de comparação com o diagrama elétrico, porém alguns novos pontos são inseridos, de

¹ pois esta entrada do CLP está recebendo nível lógico “1” de uma botoeira NF.

forma a melhorar o controle e intertravar o processo de mistura e aquecimento com outros processos controlados por outros CLP's via comunicação *Ethernet*.

Na *RUNG 1* (figura B.6) identifica-se alguns contatos que não existem no diagrama elétrico. O contato normalmente fechado %M25 é uma chave de controle do tipo DESLIGA e, como pode-se observar, ela está em série com a chave DESLIGA %I01 (referente a botoeira PB-1), ou seja, se o fluxo é impedido em qualquer desses dois contatos, desliga-se o processo de mistura. O contato %M25 é um *bit* de memória do CLP que é controlado remotamente através do *software* de supervisão e controle *Cimplicity HMI*, ou seja, ele pode ser controlado em uma tela de supervisão e controle; no caso em questão, na tela de controle geral (geral.cim / figura 5.14) e na tela de controle de mistura (misturador.cim / figura B.1), que .

Continuando na *RUNG 1*, verifica-se que o contato normalmente aberto %M24 tem a função de ligar o processo, sempre que acionado. Este contato está em paralelo ao contato normalmente aberto %I02 (referente a botoeira de partida PB-2), ou seja, pode-se ligar o processo permitindo o fluxo de energia em qualquer dos dois contatos. Da mesma forma que o contato %M25, o contato %M24 também é um *bit* de memória do CLP, controlado remotamente através do *software Cimplicity HMI*, nas mesmas telas de supervisão que controlam o *bit* %M25. Verifica-se então, que o processo pode ser ligado localmente ou remotamente através da rede de comunicação e *software* de supervisão e controle.

Ainda na *RUNG 1*, nota-se a existência do contato normalmente fechado %Q08, que é utilizado para automatizar o processo. Ele é acionado toda vez que o processo chega ao fim, de forma a recomençar automaticamente o processo, abrindo as válvulas solenóides (%Q01). Porém, para que a bobina de saída %Q01 seja energizada é necessário que o contato NF %M31 permita o fluxo de energia.

O contato %M31 refere-se ao estado do nível do líquido dos depósitos A e B; caso o nível do líquido esteja muito baixo é mandado um nível lógico "1" para a memória %M31 de forma a impedir que as válvulas solenóides se abram. O interessante é que sem a rede de comunicação não seria possível intertravar as válvulas solenóides com o processo de controle de nível, pois este é realizado por outro CLP. Logo quem escreve na memória %M31 do CLP2 é o CLP3, através de uma porta de comunicação, utilizando um canal de escrita¹, chamada *COMM_REQ*.

¹ código 2004, referente ao canal de escrita, no registro %R106 do CLP3



Na *RUNG 3*, o contato normalmente aberto %M23 é utilizado para habilitar, ou não, ligar o processo de mistura. Este contato é inicialmente levado a nível lógico “1” sempre que se liga o processo (*RUNG 22*). Sabe-se que o processo de refrigeração (CLP1) é utilizado para refrigerar o motor M1 do processo de mistura (CLP2), logo, sempre que o termostato T2, do sistema de refrigeração, estiver indicando nível alto por mais de 20 segundos, durante o tempo em que o motor M1 estiver ligado, o estado do *bit* de memória %M23 voltará a nível lógico “0” (*RUNG 21*), impedindo que o motor M1, do processo de mistura, funcione e, dessa forma, parando a etapa de mistura.

Na *RUNG 5* observa-se o seguinte, enquanto o tempo de mistura, representado pelo *bit* de memória %M03 (*RUNG 3*), não é alcançado, o *bit* %M09 mantém-se energizado, permitindo que ocorra a mistura. Quando o tempo de mistura alcança o valor pré-ajustado, %M03 é levado ao nível lógico “1”, fechando o contato NA e iniciando a próxima etapa do processo, que poderá ser o esvaziamento e transferência do líquido do reservatório ou o processo de aquecimento da mistura.

O estado do *bit* de memória %M33 é quem irá controlar qual será a próxima etapa do processo. Aquecerá a mistura (%M32) antes de esvaziar o tanque, caso %M33 esteja em nível lógico “0”, ou esvaziará o tanque (%Q06) quando seu nível lógico for “1”. O estado lógico da memória %M33 é controlado na *RUNG 25* e *RUNG 1*, e nota-se que ele será “1” quando a temperatura da mistura atingir o valor pré-ajustado, tornando-se “0” sempre que a temperatura da mistura estiver abaixo do valor pré-ajustado e as válvulas solenóides serem acionadas.

A *RUNG 6* tem a função de mandar um pulso, sempre que o processo de aquecimento tiver seu início, para o *bit* de memória %M26 que, de acordo com a *RUNG 23*, tem a função de ajustar as configurações do controlador de temperatura PID.

Os temporizadores de 100 milisegundos nas *RUNGs 10* e *11* tem a função de atrasar a alimentação das bobinas dos contatores C1 e C2, representadas em *Ladder* respectivamente por %Q02 e %Q05, evitando que ocorra um curto-circuito fase-fase, visto que o contato digital é muito mais rápido que o contato eletromecânico; outra função destes temporizadores é de permitir que as bobinas de C1 e C2 permaneçam desenergizadas durante um curto espaço de tempo para o funcionamento correto da lógica de controle, conforme explicado na lógica do diagrama elétrico.

Na *RUNG 16* tem-se o contato %I32 que indica se a rede *LAN* está funcionando corretamente. Na *RUNG 17*, o controle atua de forma a mandar um pulso para o *bit* de memória %M20 sempre que o motor M1 for ligado (em sentido horário ou anti-horário), e este *bit* de memória é utilizado para iniciar a comunicação com o CLP1, através da *COMM_REQ*, de forma a verificar o estado da temperatura da água do sistema de refrigeração.

A *RUNG 19* trata exatamente das configurações do canal de comunicação que irá ler o estado do termostato T2 do sistema de refrigeração¹ (CLP1), e como pode-se observar ela executará a configuração do canal de comunicação sempre que a *LAN* estiver funcionando corretamente e %M20 receber um pulso. A *RUNG 20* trata especificadamente do bloco de função referente ao *COMM_REQ*, que será energizado nas mesmas condições da *RUNG 19*.

O *bit* de memória %M22 recebe, através da *COMM_REQ*, a informação referente ao estado do termostato T2 do sistema de refrigeração, controlado pelo CLP1. A *RUNG 21* descreve o tempo em que o termostato T2, do sistema de refrigeração, permanece em nível lógico “1”.

Nas *RUNG 23* e *24* estão presentes a configuração e o bloco de função do controlador PID, que controlará a temperatura da mistura. De acordo com estas *RUNGs*, o controlador PID comparará o valor da entrada analógica, armazenada na memória %AI01, com a constante 5000, que representa, após conversão de unidade, aproximadamente 45 °C. A variável de controle será atualizada a cada 1seg, armazenada no registro de memória %AQ01 e transferida, após uma conversão D/A, em Volts para a saída do CLP1, que atuará na resistência responsável pelo aquecimento da mistura.

Embora a variável de controle funcione como o esforço do controle PID para alcançar o regime em 5000, este nunca será alcançado, visto que na *RUNG 25* é inserido um comparador de temperatura em 4000, que desligará o PID sempre que a temperatura (%AI01) chegar a este valor. A razão pela qual adota-se tal procedimento foi que, a variável do processo sofre um *overshooting* quando se utiliza uma malha de controle PID e como o atuador para o aquecimento é somente uma resistência elétrica, não consegue-se diminuir a temperatura da mistura através do controlador quando ela ultrapassar o valor desejado.

¹ Este estado é descrito pelo *bit* de memória %I03 do CLP1.



Assim, a temperatura da mistura iria diminuir apenas com a troca de calor com o ambiente, ligando novamente o PID quando a temperatura (%AI01) for menor que 5000; sendo, portanto, um evento muito demorado que não torna viável utilizá-lo. Uma boa sugestão para tal planta é a utilização de um controlador proporcional, que não causa um *overshooting*, porém que decorre num pequeno erro em regime permanente.

As *RUNGS* 26, 27 e 28 são utilizadas para enviar ao registro %R500 algumas situações do processo e este registro será lido pelo CLP1, para ser usado no sistema de sinalização via pêndulo invertido. Na *RUNG* 26 sempre que o processo de mistura for ligado, o valor -1600 será movido para o registro %R500. Na *RUNG* 27, quando as válvulas solenóides estiverem abertas ou quando o processo de aquecimento estiver funcionando, o valor 0 (zero) será movido para o registro %R500. Finalmente, na *RUNG* 28 quando a bomba M2, que transfere a mistura aquecida para o depósito C, for ligada, o valor +1600 é movido para o registro %R500. Logo, verificando o estado do registro %R500 pode-se ter uma idéia do que está sendo executado na planta em questão.

A última *RUNG* é utilizada para mover o valor 0 (zero), para a memória %AQ01, responsável pela variável de controle do aquecedor, sempre que a etapa de aquecimento não estiver em funcionamento.

B.2 CONTROLADOR DE NÍVEL DE LÍQUIDO

O controlador de nível é utilizado para manter o nível do líquido, dos depósitos A e B (figura B.1), num valor apropriado para a mistura ocorrer corretamente. Por redundância, será apresentado apenas o controle de um depósito, assim, o tanque da figura B.10 representa os dois depósitos da figura B.1.

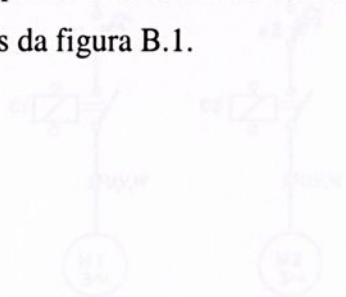


Figura B.11 - diagrama de força motriz dos motores do controlador de nível.

O controlador de nível (CLP3) deverá informar ao processo de mistura (CLP2), através da rede de comunicação, se o nível do líquido no depósito está baixo ou em operação normal; estando o nível do depósito num valor baixo, o controlador informa o CLP2,

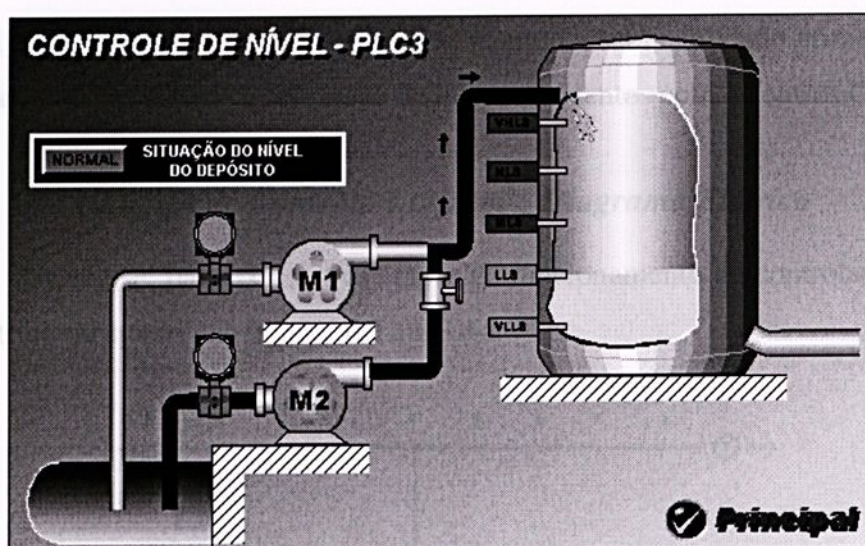


Figura B.10 - tela de supervisão e controle do processo de controle de nível.

Para a realização deste controle utiliza-se dois motores, que funcionarão dependendo do fluxo de líquido exigido. Cinco sensores são utilizados para verificar o nível do líquido no tanque e outros dois sensores são colocados na entrada das bombas (M1 e M2) para verificar se está ocorrendo fluxo de líquido ou se a bomba está operando a vazio.

As duas bombas partem na configuração estrela sem necessidade de mudança para configuração triângulo. O diagrama unifilar de força para ligação dos motores é mostrado na figura B.11.

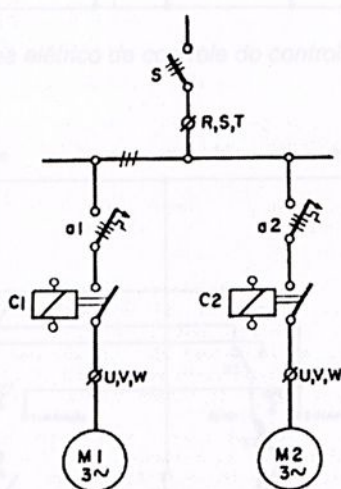


Figura B.11 - diagrama de força unifilar dos motores do controlador de nível.

O controlador de nível (CLP3) deverá informar ao processo de mistura (CLP2), através da rede de comunicação, se o nível do líquido no depósito está baixo ou em operação normal; estando o nível do depósito num valor baixo, o controle informará o CLP2,

que impedirá a abertura das válvulas solenóides. Contudo, será explicado primeiro o processo de controle de nível em si, sem a utilização de intertravamentos com os outros CLPs.

B2.1 Lógica de Controle Utilizando Diagrama Elétrico

Utiliza-se, para explicar as etapas de funcionamento do controlador de nível, o diagrama elétrico mostrado nas figuras B.12 à B.14.

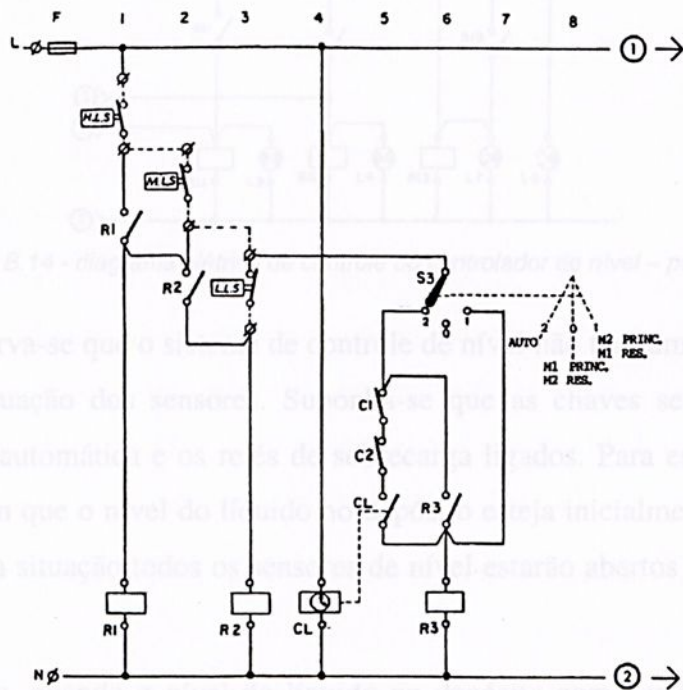


Figura B.12 - diagrama elétrico de controle do controlador de nível – parte 1/3.

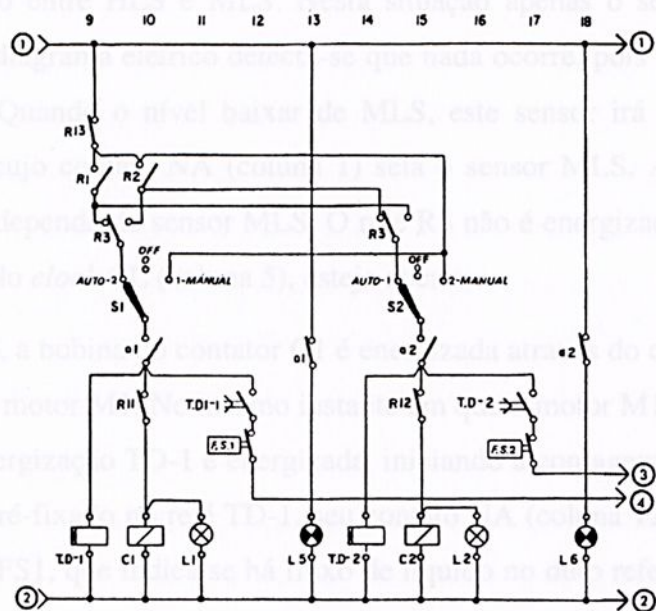


Figura B.13 - diagrama elétrico de controle do controlador de nível – parte 2/3.

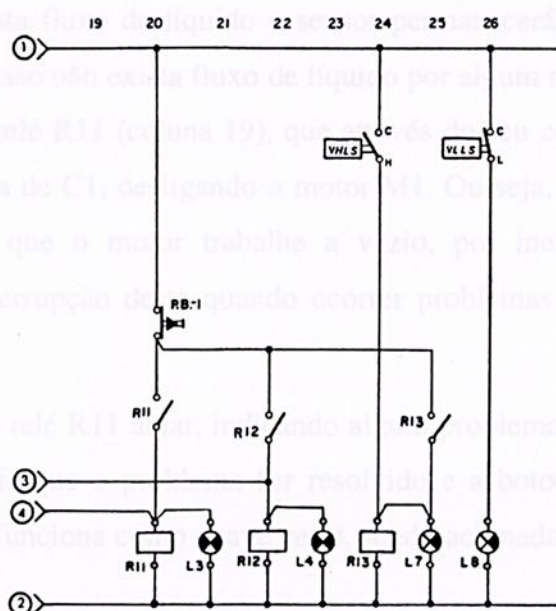


Figura B.14 - diagrama elétrico de controle do controlador de nível – parte 3/3.

Observa-se que o sistema de controle de nível não tem uma chave geral, logo a análise parte da situação dos sensores. Suponha-se que as chaves seletoras S1, S2 e S3 estejam na posição automática e os relés de sobrecarga ligados. Para estudar esse processo, considera-se também que o nível do líquido no depósito esteja inicialmente entre os sensores VHLS e HLS. Nesta situação todos os sensores de nível estarão abertos e os motores estarão desligados.

Assim, quando o nível do líquido no depósito começar a baixar, os sensores atuarão, informando ao controle a situação do nível. A análise inicia supondo primeiro que o nível baixou, ficando entre HLS e MLS. Nesta situação apenas o sensor HLS é fechado, porém analisando o diagrama elétrico detecta-se que nada ocorre, pois o sensor MLS (coluna 2) continua aberto. Quando o nível baixar de MLS, este sensor irá fechar e energizará a bobina do relé R1, cujo contato NA (coluna 1) sela o sensor MLS. A partir deste ponto a lógica de controle independe do sensor MLS. O relé R3 não é energizado se considerar que o contato auxiliar NA do *clock* CL (coluna 5), esteja aberto.

Assim, a bobina do contator C1 é energizada através do contato NA do relé R1 (coluna 9), ligando o motor M1. No mesmo instante em que o motor M1 é ligado, a bobina do relé de retardo na energização TD-1 é energizada, iniciando a contagem do tempo pré-fixado. Decorrido o tempo pré-fixado no relé TD-1, seu contato NA (coluna 12) fechará, verificando a situação do sensor FS1, que indica se há fluxo de líquido no duto referente à bomba (motor M1).

Caso exista fluxo de líquido o sensor permanecerá aberto e o motor M1 em funcionamento, porém caso não exista fluxo de líquido por algum motivo, o sensor FS1 fecha e energiza a bobina do relé R11 (coluna 19), que através do seu contato auxiliar NF (coluna 10) desenergiza a bobina de C1, desligando o motor M1. Ou seja, o sensor FS1 é um sensor de proteção que evita que o motor trabalhe a vazio, por inexistência de líquido para transferência ou por interrupção deste quando ocorrer problemas (quebrar, entupir, etc) no duto.

Quando o relé R11 atuar, indicando algum problema de fluxo, o sistema só irá voltar a funcionar depois que o problema for resolvido e a botoeira PB-1 (coluna 20) for acionada. Esta botoeira funciona como chave *reset*, sendo acionada sempre que um problema do sistema for corrigido.

Considera-se que o fluxo do líquido fornecido pela bomba M1 é suficiente para aumentar o nível do líquido no reservatório. Assim, quando o nível ultrapassa novamente o sensor MLS, este sensor se abre, mas nada ocorre, pois o sensor está selado pelo contato auxiliar NA do relé R1 (coluna 1). Porém, quando o nível do líquido ultrapassa o sensor HLS, este sensor se abre, desligando todo processo de controle de nível.

Supõe-se agora que o nível do líquido voltou a baixar depois que o *clock* CL contou o seu tempo pré-fixado, fechando o seu contato NA (coluna 5). Desta vez, quando o nível do líquido baixar de MLS o controle atuará de forma a energizar a bobina do relé R3 mudando o estado dos seus contatos auxiliares. Assim, verificando o controle nota-se que a bobina do contator C2 é energizada e assim, o motor M2 é ligado.

Logo, percebe-se que a função do *clock* é de intercalar o funcionamento entre os motores, evitando que sempre o mesmo motor entre em funcionamento, causando maior depreciação sobre o mesmo. Da mesma forma que FS1, agora FS2 é quem verifica o fluxo do líquido causado pela bomba (motor M2). Supondo que ocorra algum problema no fluxo do líquido, a bobina do relé R12 é energizada, desenergizando, através do seu contato NF (coluna 15), a bobina do contator C2 e juntamente com ela o motor M2. Detectado e solucionado o problema de fluxo, pressiona-se a botoeira PB-1 que religa o motor M2.

Passa-se agora a analisar uma nova possibilidade. Supõe-se que o fluxo de líquido, fornecido pela bomba, não seja suficiente para elevar o nível do líquido no depósito. Assim o nível continua caindo até baixar de LLS. Nesta situação o sensor LLS fecha-se

energizando a bobina do relé R2. Assim pode-se verificar, na coluna 10, que o contato NA de R2 fecha-se e liga o outro motor, não importando qual seja a situação do relé R3.

Nesta situação os dois motores estarão ligados aumentando o fluxo de líquido para o depósito, com o intuito de elevar o nível do líquido. Se os dois motores forem suficientes para elevar o nível do líquido no depósito eles serão desligados quando o nível atingir HLS.

Os sensores VHLS e VLLS são sensores que indicam problemas no controle de líquido. Supondo que, mesmo com os dois motores ligados, o fluxo não seja suficiente para elevar o nível do líquido, o sensor VLLS acusará nível muito baixo e será ligada uma lâmpada (L8) de alarme. Caso o nível do líquido atinja um valor muito alto, devida alguma falha no controle dos motores, o sensor VHLS fechará energizando a bobina do relé R13, ligando a lâmpada de sinalização de falha (L7) e desligando os motores através do contato auxiliar NF de R13 (coluna 9). Ao solucionar o problema referente ao sensor VHLS, basta pressionar a botoeira PB-1 para o sistema voltar a funcionar corretamente.

B2.2 Lógica de Controle Utilizando Diagrama Ladder

Como descrito, o controlador de nível (CLP3) tem o objetivo de controlar o nível do líquido, deixando-o em determinada faixa de valores. Ele controla o nível do líquido nos depósitos A e B do processo de mistura, logo o CLP3 deverá informar ao CLP2 a situação do nível, e isso é feito através do canal de comunicação utilizando o *COMM_REQ*.

A diferença existente entre o diagrama elétrico e o diagrama *Ladder*, para esse controle, são3: i) maneira de configurar o *clock*; ii) inserção da comunicação com o CLP2, e iii) inserção de pontos para controle remoto através do *software Cimplicity HMI*.

A seguir apresenta-se, nas figuras B.15 e B.16, o diagrama *Ladder* para esse processo e, logo a seguir, é descrito as principais *RUNGS* deste diagrama.

Figura B.15- diagrama Ladder do controlador de nível - parte 1/2



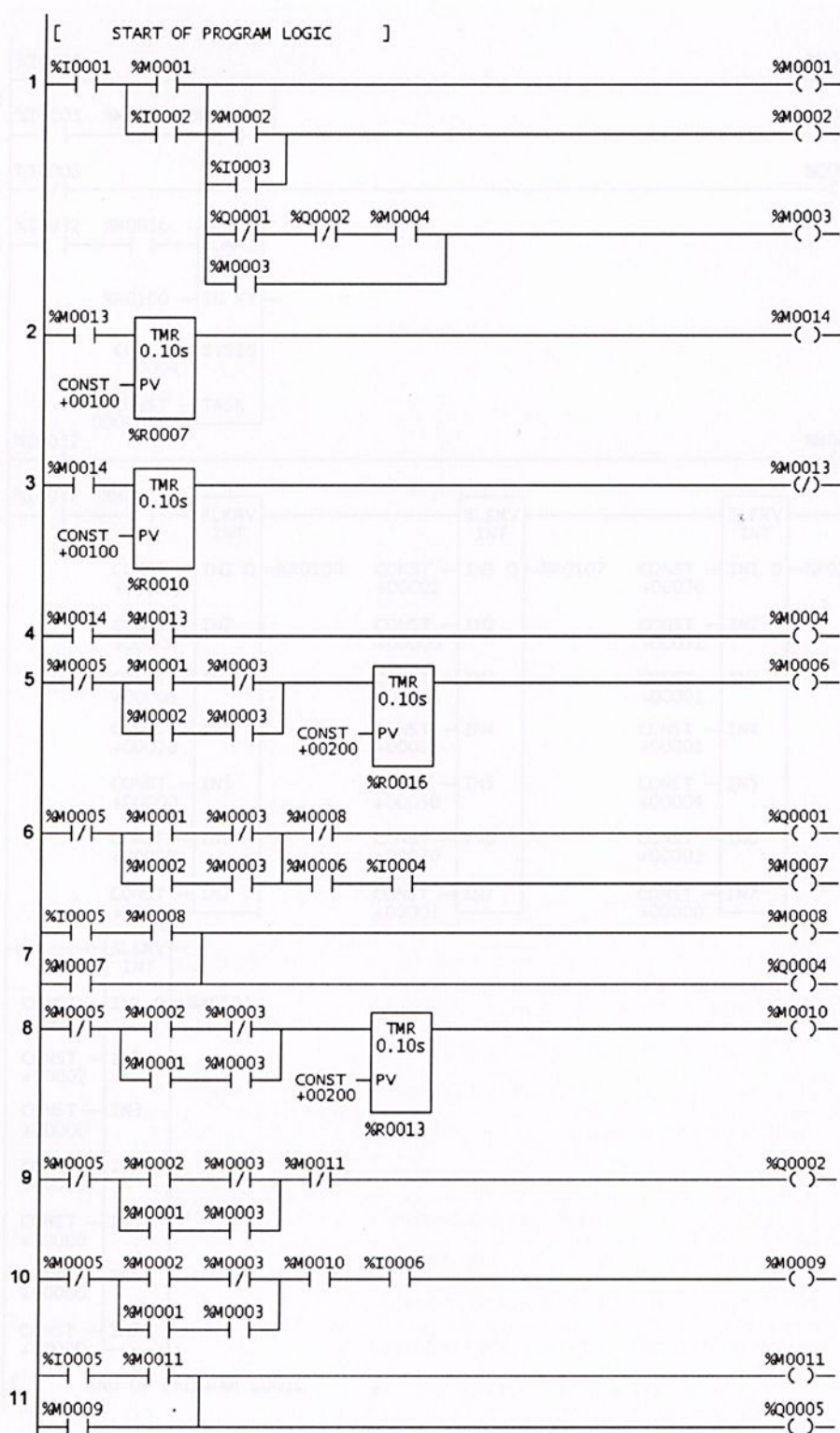


Figura B.15- diagrama Ladder do cotrolador de nível – parte 1/2.

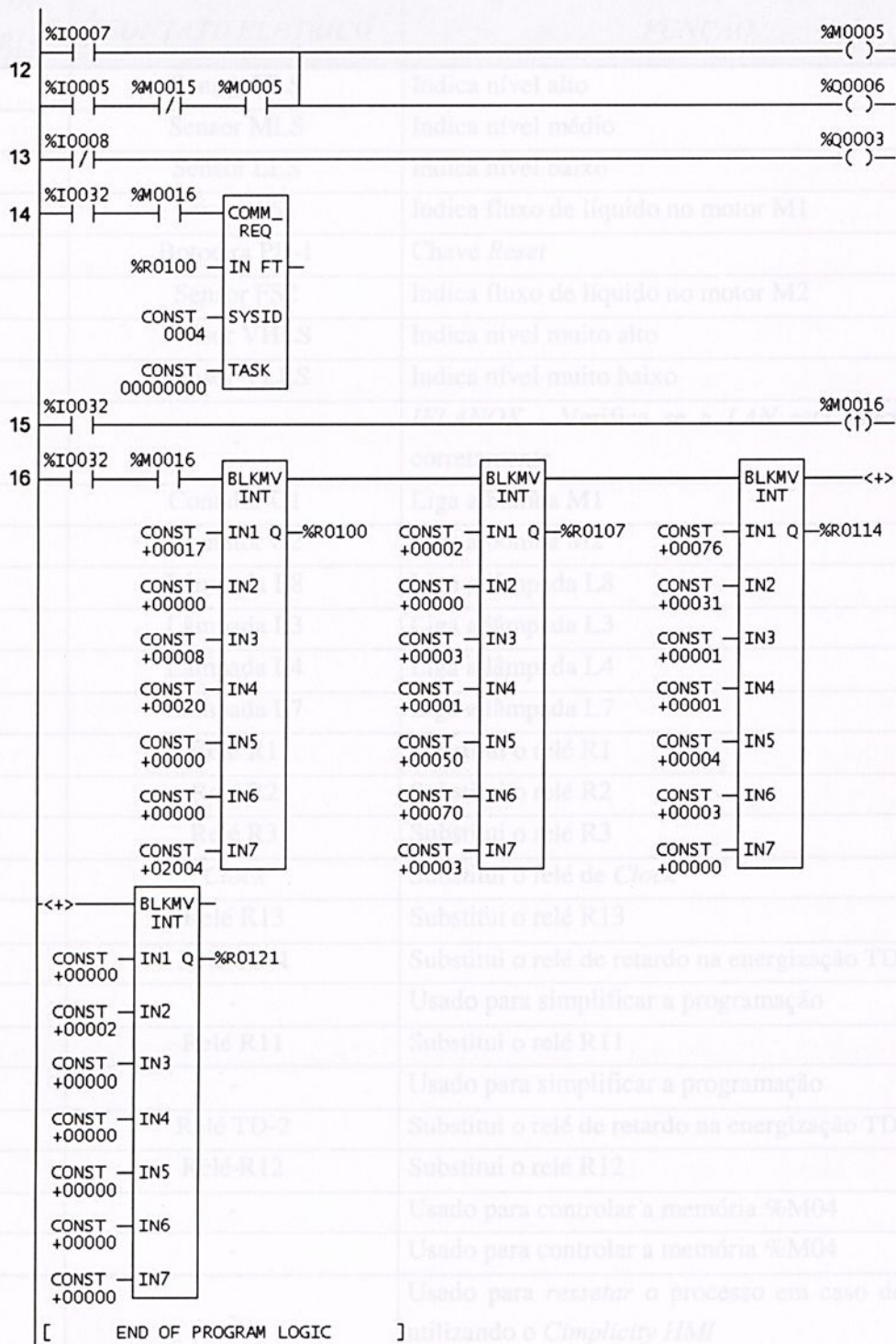


Figura B.16 - diagrama Ladder do cotrolador de nível – parte 2/2.

A tabela B.4, apresenta todos locais de memória, utilizados na lógica de programação *Ladder* do controlador de nível, seu contato elétrico referente (se existir) e a sua função.

LOCAL DE MEMÓRIA	CONTATO ELÉTRICO	FUNÇÃO
%I01	Sensor HLS	Indica nível alto
%I02	Sensor MLS	Indica nível médio
%I03	Sensor LLS	Indica nível baixo
%I04	Sensor FS1	Indica fluxo de líquido no motor M1
%I05	Botoeira PB-1	Chave <i>Reset</i>
%I06	Sensor FS2	Indica fluxo de líquido no motor M2
%I07	Sensor VHLS	Indica nível muito alto
%I08	Sensor VLLS	Indica nível muito baixo
%I32	-	<i>IFLANOK</i> - Verifica se a <i>LAN</i> está funcionando corretamente
%Q01	Contator C1	Liga a bomba M1
%Q02	Contator C2	Liga a bomba M2
%Q03	Lâmpada L8	Liga a lâmpada L8
%Q04	Lâmpada L3	Liga a lâmpada L3
%Q05	Lâmpada L4	Liga a lâmpada L4
%Q06	Lâmpada L7	Liga a lâmpada L7
%M01	Relé R1	Substitui o relé R1
%M02	Relé R2	Substitui o relé R2
%M03	Relé R3	Substitui o relé R3
%M04	<i>Clock</i>	Substitui o relé de <i>Clock</i>
%M05	Relé R13	Substitui o relé R13
%M06	Relé TD-1	Substitui o relé de retardo na energização TD-1
%M07	-	Usado para simplificar a programação
%M08	Relé R11	Substitui o relé R11
%M09	-	Usado para simplificar a programação
%M10	Relé TD-2	Substitui o relé de retardo na energização TD-2
%M11	Relé R12	Substitui o relé R12
%M13	-	Usado para controlar a memória %M04
%M14	-	Usado para controlar a memória %M04
%M15	-	Usado para <i>resetar</i> o processo em caso de falha, utilizando o <i>Cimplicity HMI</i>
%M16	-	Usado para estabelecer a comunicação
%R07	-	Registro de temporizador
%R10	-	Registro de temporizador
%R13	-	Registro de temporizador
%R16	-	Registro de temporizador
%R100- %R122	-	Configura a <i>COMM_REQ</i>

Tabela B.4 - locais de memória utilizados na lógica de programação.



Na *RUNG 1* o diagrama *Ladder* é similar ao diagrama elétrico (figuras B.12 à B.14), não havendo modificações na lógica. O contato do *clock* é representado pela memória %M04.

As *RUNGS 2,3 e 4* tem a função de criar o pulso de *clock* que controla o estado do *bit* de memória %M04. Esta memória fica alternando seu estado lógico, permanecendo em cada estado durante 10 segundos.

Até a *RUNG 13* o diagrama *Ladder* é de fácil análise, devido a semelhança com o diagrama elétrico, mudando-se apenas as simbologias gráficas adotadas. A única diferença está na inserção do contato normalmente fechado %M15 na *RUNG 12*, que tem a mesma função da botoeira PB-1 (%I05), porém controlado remotamente pelo *software* de supervisão e controle *Cimplicity HMI*.

Na *RUNG 14* insere-se o bloco de comunicação *COMM_REQ*, que será ligado quando a rede *LAN* estiver funcionando corretamente (%I32) e quando for dado um pulso através da memória %M16. Na *RUNG 15* é criado o pulso de memória %M16, que ocorre quando a rede *LAN* estiver funcionando sem problemas (%I32). Finalmente a última *RUNG* tem a função de configurar o canal de comunicação entre o CLP3 e o CLP2. O *COMM_REQ* foi configurado para escrever¹ o estado do *bit* de memória de entrada %I03 (sensor LLS) no *bit* da memória %M31 do CLP2, que controlará a abertura ou não das válvulas solenóides no processo de mistura.

B.3 SISTEMA DE REFRIGERAÇÃO E SINALIZAÇÃO VIA PÊNDULO INVERTIDO

O sistema de refrigeração (CLP1) tem uma função muito importante na simulação laboratorial da pequena indústria, apresentada aqui como proposta final para o curso, utilizando o laboratório de ensino implantado por consequência de pesquisas em automação. Este sistema é responsável pela refrigeração do motor M1 do processo de mistura (CLP2).

Como o motor M1 parte em carga várias vezes, invertendo o sentido de rotação do seu eixo, ele absorve muita corrente, podendo esquentar muito, causando danos em seus enrolamentos e diminuindo sua vida útil. Pensando nisto, o sistema de refrigeração foi projetado para manter uma água gelada circulando numa camisa que reveste o motor M1, de

¹ código 2004, referente ao canal de escrita, no registro %R106 do CLP3



forma a manter sua temperatura num valor suficientemente baixo, evitando assim maiores problemas.

Para analisar o controle do processo de refrigeração, adota-se o mesmo método utilizado nos processos de mistura e de controle de nível, ou seja, primeiramente analisa-se o processo funcionando independente dos demais processos, utilizando um diagrama elétrico, e por fim, apresenta-se o diagrama *Ladder*, correspondente ao elétrico, inserindo as novas funções, intertravamentos, e o sistema de sinalização via pêndulo invertido.

A figura B.17 apresenta o sistema de refrigeração e pode-se ver nela que serão necessários quatro motores: um compressor, um ventilador e duas bombas (M1 e M2). Presentes também no sistema dois sensores de temperatura (termostatos T1 e T2), dois sensores de pressão (pressostatos HP e LP) e uma válvula solenóide.

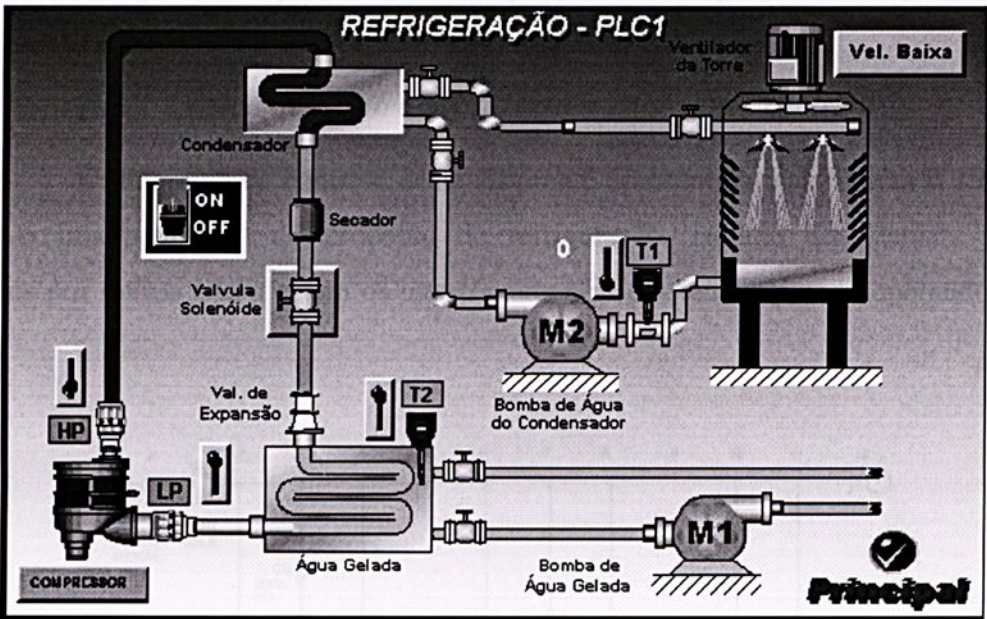


Figura B.17 - tela de supervisão e controle do sistema de refrigeração.

As duas bombas M1 e M2 partem de forma direta, enquanto que o compressor (M4) parte na configuração estrela e depois passa para triângulo. O ventilador opera com duas frequências de rotações diferentes, dependentes do número de pólos da máquina. A figura B.18 mostra o diagrama elétrico de força que, embora não seja exatamente este o diagrama de força a ser implementado na simulação laboratorial¹, serve como referência para descrição do processo.

¹ O motor tipo “Dhahlander” (ventilador/M3) será substituído por um inversor de frequência, de forma a obter 2 velocidades diferentes, dependendo apenas do controle implementado.



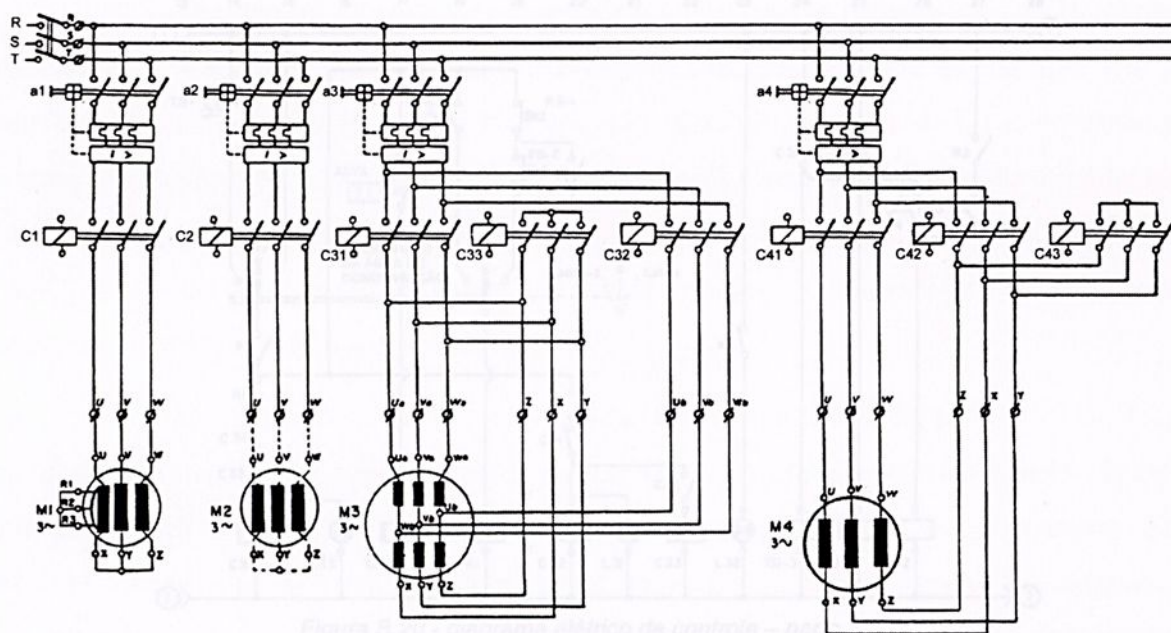


Figura B.18 - diagrama de força do sistema de refrigeração sem utilização de inversor de frequência.

B.3.1 Lógica de Controle Utilizando Diagrama Elétrico

O diagrama de controle deste sistema é apresentado nas figuras B.19 à B.21 e descreve o funcionamento do processo de refrigeração, isolado dos demais processos controlados por outros CLPs e sem os intertravamentos existentes quando operando em rede de CLPs.

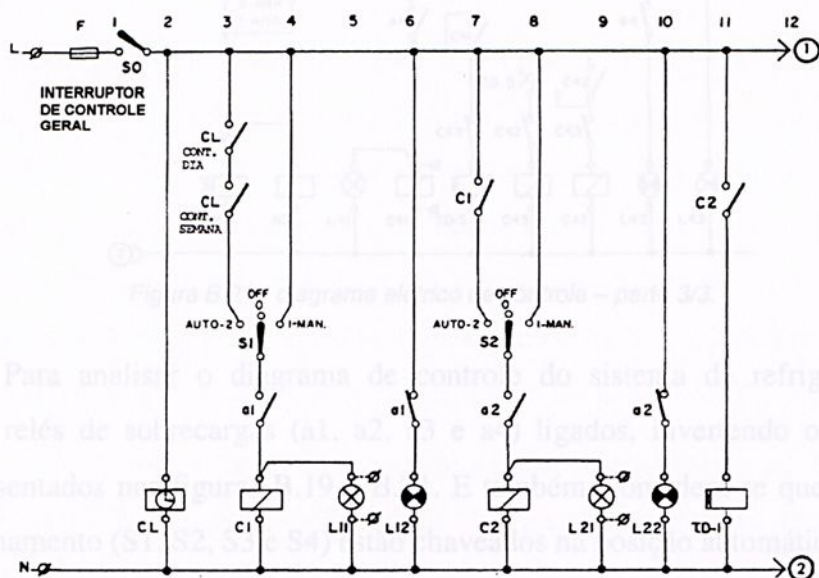


Figura B.19 - diagrama elétrico de controle - parte 1/3.

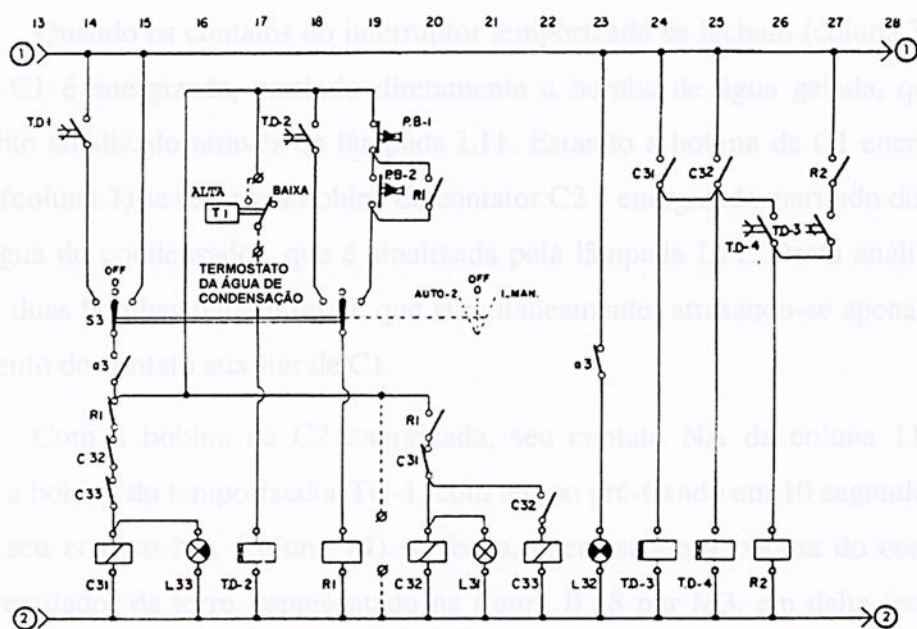


Figura B.20 - diagrama elétrico de controle – parte 2/3.

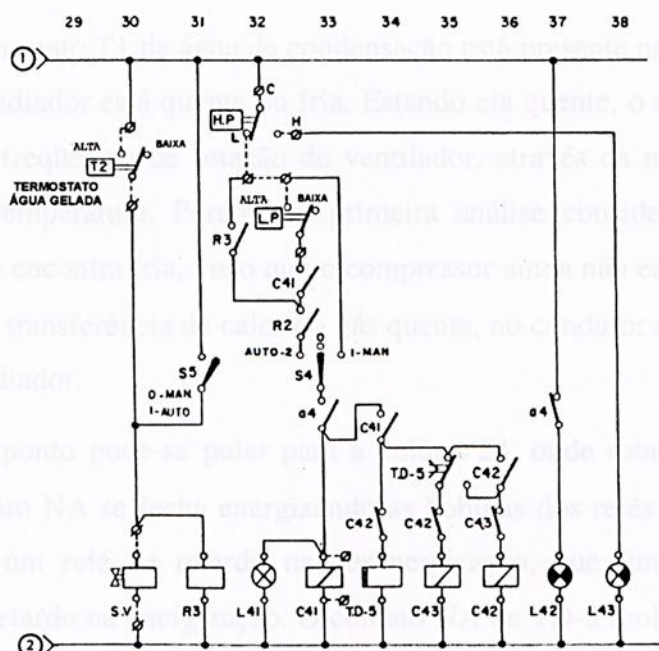


Figura B.21 - diagrama elétrico de controle – parte 3/3.

Para analisar o diagrama de controle do sistema de refrigeração, deve-se considerar os relés de sobrecargas (a1, a2, a3 e a4) ligados, invertendo o estado de seus contatos, apresentados nas figuras B.19 à B.21. E também, considera-se que os seletores de tipo de funcionamento (S1, S2, S3 e S4) estão chaveados na posição automática.

Na 1ª coluna tem-se a chave geral do sistema de controle. Estando ela ligada, o sistema opera automaticamente através de um interruptor temporizado de dia e semana.

Quando os contatos do interruptor temporizado se fecham (coluna 3), a bobina do contator C1 é energizada, partindo diretamente a bomba de água gelada, que tem seu funcionamento sinalizado através da lâmpada L11. Estando a bobina de C1 energizada, seu contato NA (coluna 7) se fecha e a bobina do contator C2 é energizada, partindo diretamente a bomba de água do condensador, que é sinalizada pela lâmpada L21. Desta análise, pode-se dizer que as duas bombas partem quase que simultaneamente, atrasando-se apenas do tempo de chaveamento do contato auxiliar de C1.

Com a bobina de C2 energizada, seu contato NA da coluna 11 se fecha, energizando a bobina do temporizador TD-1, com tempo pré-fixado em 10 segundos. Passado este tempo, seu contato NA (coluna 14) se fecha, energizando a bobina do contator C31, partindo o ventilador da torre, representado na figura B.18 por M3, em delta (configuração esta que, nesta máquina, produz 4 pólos¹); o funcionamento, deste motor, em baixa velocidade é sinalizado pela lâmpada L33.

O termostato T1 da água de condensação está presente na coluna 17, indicando se a água vinda do radiador está quente ou fria. Estando ela quente, o controle deve atuar de forma a aumentar a frequência de rotação do ventilador, através da mudança para 2 pólos, para diminuir esta temperatura. Porém em primeira análise considera-se que a água do condensador ainda se encontra fria, visto que o compressor ainda não está em funcionamento, ou seja, ainda não há transferência de calor do gás quente, no condutor de refrigeração, com a água fria vinda do radiador.

Deste ponto pode-se pular para a coluna 24, onde estando a bobina de C31 energizada, seu contato NA se fecha energizando as bobinas dos relés temporizados TD-3 e TD-4. Sendo TD-3 um relé de retardo na desenergização, que funciona completamente contrário ao relé de retardo na energização. O contato NA de TD-3 (coluna 27) se fecha, mas nada ocorre, visto que o contato auxiliar de R2, também na coluna 27, e o contato de TD-4, ambos NA, não estão fechados. Decorrido o tempo pré-fixado em TD-4, seu contato se fecha energizando a bobina de R2 e fechando o contato NA de R2 da coluna 27.

Como o compressor ainda não foi ligado, a temperatura no depósito de água gelada está alta, logo o termostato T2 (coluna 30) acusa temperatura alta, energizando a válvula solenóide e a bobina do relé R3.

¹ O número de pólos é inversamente proporcional a frequência de rotação.



O pressostato HP, da coluna 32, a princípio acusa nível baixo, sendo que, quando atingir nível alto o controle atua de forma a desligar o compressor, evitando que por problemas de pressão possa estourar os dutos onde o gás flui; a lâmpada L43, indica defeito na pressão de saída do compressor.

Com o sensor HP em nível baixo e as bobinas de R2 e R3 energizadas, a bobina do contator C41 será energizada, fechando seu contato NA (coluna 34), energizando também a bobina do relé de retardo na energização TD-5 e o contator C43. Analisando o circuito de força, na figura B.18, observa-se que, quando os contadores C41 e C43 estão energizados o compressor é ligado na configuração estrela, permanecendo nesta até o temporizador TD-5 terminar sua contagem, mudando a posição de seu contato auxiliar (coluna 35) de NF para NA, desenergizando C43 e energizando C42, ou seja, passando da configuração estrela para a configuração triângulo. Enquanto este motor estiver funcionando, a lâmpada L41 sinalizará seu funcionamento.

O sistema já está em funcionamento completo, logo já existe a troca de calor entre o gás do sistema de refrigeração e a água do condensador. Supõe-se que, em determinado instante, a água do condensador esquentou de forma que o radiador não conseguiu esfriá-la novamente abaixo da temperatura limite, assim, o termostato T1 (coluna 17) acusa nível alto, fechando seu contato e energizando o temporizador TD-2 que, após determinado tempo, fecha seu contato na coluna 18 e energiza a bobina do relé R1, desenergizando a bobina do contator C31 e energizando as bobinas de C32 e C33.

Nesta configuração, C32 e C33 energizado e C31 desenergizado, as bobinas do motor do ventilador M3 são ligadas em dupla estrela, produzindo 2 pólos, aumentando assim a frequência de rotação do ventilador em duas vezes, e conseqüentemente, a capacidade de resfriamento da água do condensador.

Na comutação do ventilador operando em baixa velocidade para alta velocidade e vice-versa, é importante verificar um pequeno, porém significativo detalhe. Na comutação existe um pequeno instante, na casa de milissegundos, em que os contatos NA de C31 e C32 (colunas 24 e 25) estarão abertos, desta forma, as bobinas dos relés de tempo TD-3 e TD-4 são desenergizadas, logo o contato de TD-4 volta a se abrir. Não fosse o relé de retardo na desenergização TD-3, o relé R2 seria desenergizado, desligando o compressor.

Supõe-se agora que a temperatura do depósito de água gelada baixou até a temperatura desejada. Logo o termostato T2 abre seu contato (coluna 30), fecha a válvula



solenóide e desenergiza o relé R3. Enquanto o pressostato LP (medidor de baixa pressão colocado na entrada do compressor) estiver acusando alta pressão o compressor não é desligado, logo ele continua comprimindo gás na sua saída e retirando o gás remanescente na sua entrada, até que LP acuse nível baixo, desligando o compressor.

As lâmpadas L12, L22, L32 e L42 são utilizadas para sinalizar sobrecarga nos motores M1, M2, M3 e M4, respectivamente.

B.3.2 Lógica de Controle Utilizando Diagrama Ladder

Como descrito, o motor *Dhahlander* é substituído por um motor simples, controlado por um inversor de frequência, resultando num novo diagrama de força, conforme descrito na figura B.22.

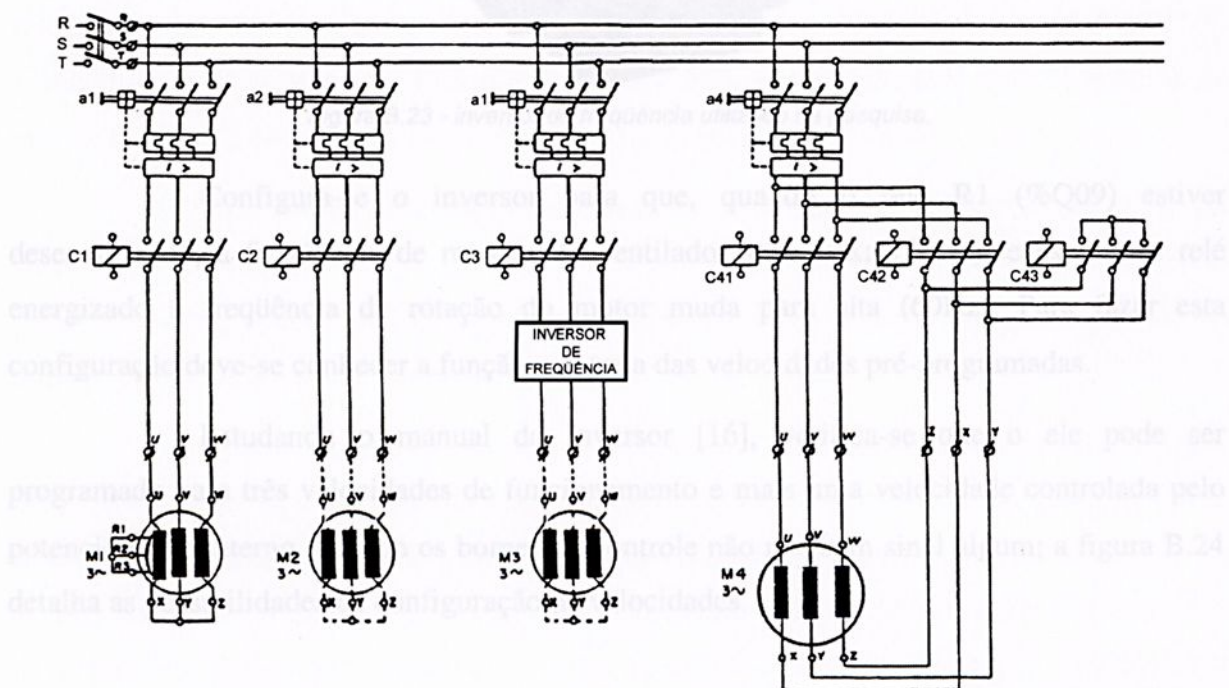


Figura B.22 - diagrama de força do sistema de refrigeração com utilização de inversor de frequência.

Este é o motor usado como ventilador da torre (M3), que opera em duas velocidades diferentes. O diagrama de controle, apresentado nas figuras B.19 à B.21, acabou sendo simplificado, visto que não é necessário mais controlar os contatores C31, C32 e C33. Para o controle de velocidade basta ligar o inversor de frequência, ou seja, energizar a bobina do contator C3, e mandar pulsos de 24Vcc para o borne de controle do inversor, de acordo com a velocidade desejada.

O inversor de frequência, utilizado para controlar o ventilador foi o modelo trifásico *FVR-C9S GE Fuji Electric*, que é de fácil programação. Ele possui 34 funções de programação, onde pode-se escolher, por exemplo: torque, tempo de aceleração e desaceleração, frequência máxima de rotação, velocidades pré-programadas, etc.



Figura B.23 - inversor de frequência utilizado na pesquisa.

Configura-se o inversor para que, quando o relé R1 (%Q09) estiver desenergizado, a frequência de rotação do ventilador seja baixa (20Hz), e estando o relé energizado a frequência de rotação do motor muda para alta (60Hz). Para fazer esta configuração deve-se conhecer a função que trata das velocidades pré-programadas.

Estudando o manual do inversor [16], verifica-se que o ele pode ser programado para três velocidades de funcionamento e mais uma velocidade controlada pelo potenciômetro interno, quando os bornes de controle não recebem sinal algum; a figura B.24 detalha as possibilidades de configuração de velocidades.

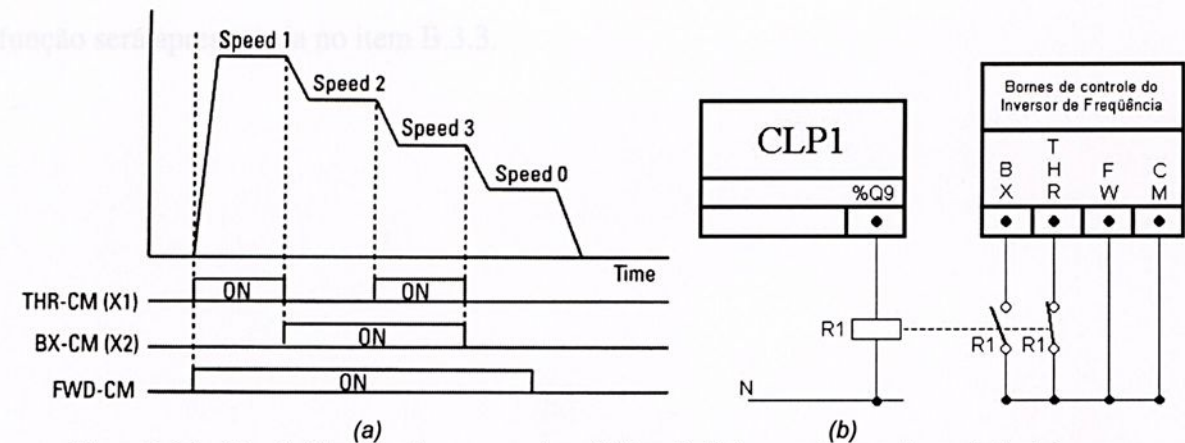


Figura B.24 - (a) velocidades pré-programadas, (b) ligação física nos bornes de controle do inversor

A figura B.24 indica que estando X1 em nível lógico “1” e X2 em “0” o inversor é programado para gerar uma frequência de velocidade *Speed 1*, quando X1 está em “0” e X2 em “1” ele gera a velocidade *Speed 2*, quando X1 e X2 estão em “1” a velocidade é *Speed 3*, e finalmente, quando X1 e X2 estão em “0” o inversor gera a frequência de rotação *Speed 0*.

Armazena-se as velocidades pré-programadas nas funções F29 (*Speed 1*), F30 (*Speed 2*) e F31 (*Speed 3*), porém, conforme pode-se observar, no esboço da ligação elétrica da figura B.24, nunca tem-se, ao mesmo tempo, nível lógico “0” ou “1” em X1 e X2, mesmo porque, para este ventilador, não é necessário programar mais de duas velocidades, assim sendo, não importa o valor da velocidade ajustada na função F31.

Outro detalhe importante é que para utilizar o ajuste de velocidade através de um sinal externo, como no caso em questão, deve-se ajustar a função F02, do inversor de frequência, para 1, habilitando o uso de sinal externo e desabilitando o botão *RUN / STOP*. Também deve-se ajustar para 1 as funções F27 e F28, disponibilizando respectivamente os bornes THR e BX para serem usados no ajuste de frequência de rotação.

Uma última observação é que o inversor de frequência controla um motor de indução, logo a frequência de rotação que o inversor gera é referente ao campo girante, sendo assim, o motor gira com uma frequência de rotação um pouco inferior ao valor mostrado no *display*, devido ao escorregamento.

Apesar do uso do inversor de frequência para controlar o ventilador, o que simplifica o controle, o diagrama de controle, escrito em *Ladder*, não está mais simples que o apresentado no diagrama elétrico, pois é adicionado a este os intertravamentos referentes ao funcionamento da indústria como um todo, e também, porque são inseridos novos pontos referentes às informações adquiridas e transmitidas ao sistema do pêndulo invertido, cuja função será apresentada no item B.3.3.



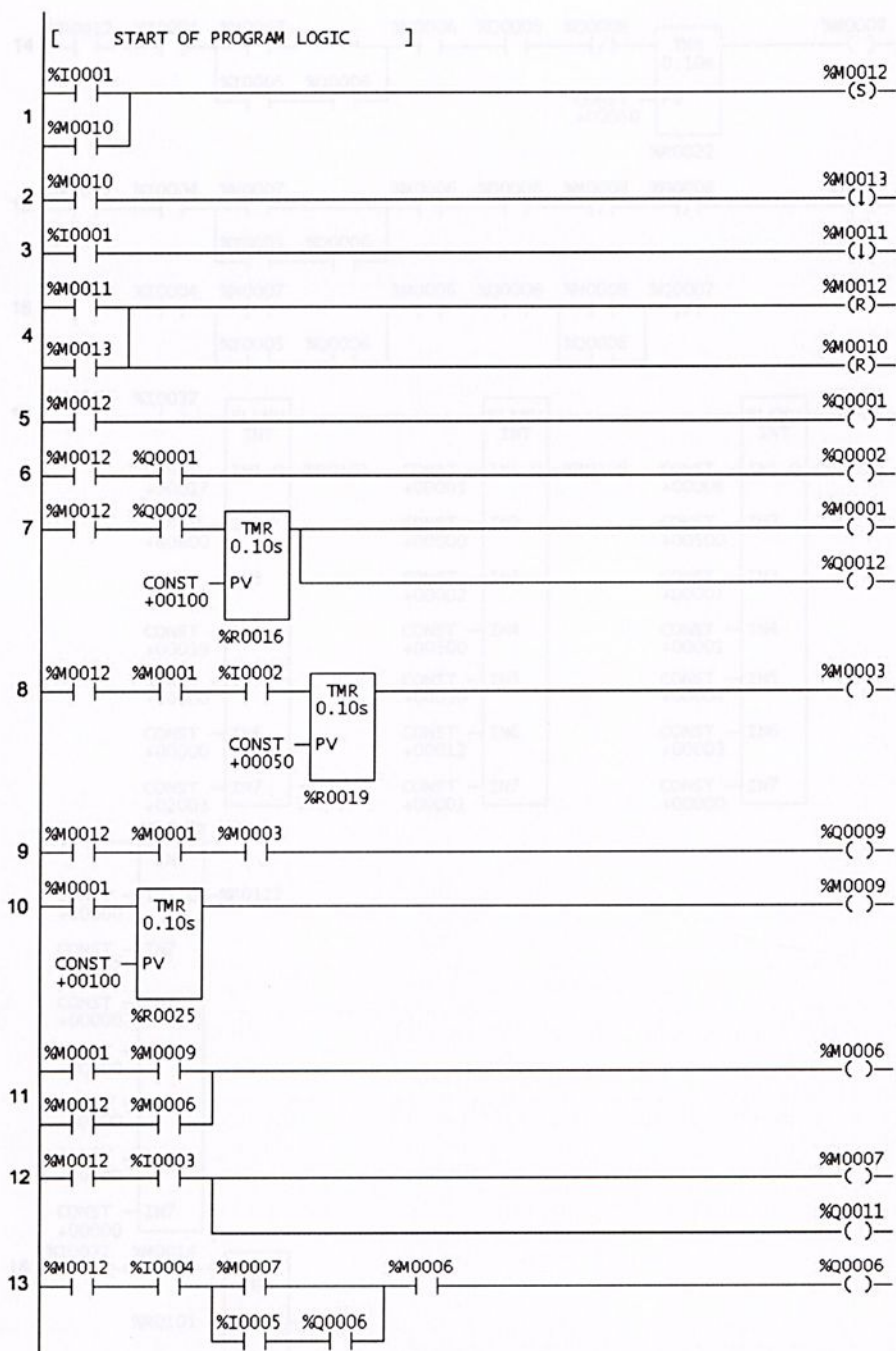


Figura B.25 - diagrama Ladder do sistema de refrigeração e sinalização via pêndulo invertido – parte 1/2.

Figura B.26 - diagrama Ladder do sistema de refrigeração e sinalização via pêndulo invertido – parte 2/2.

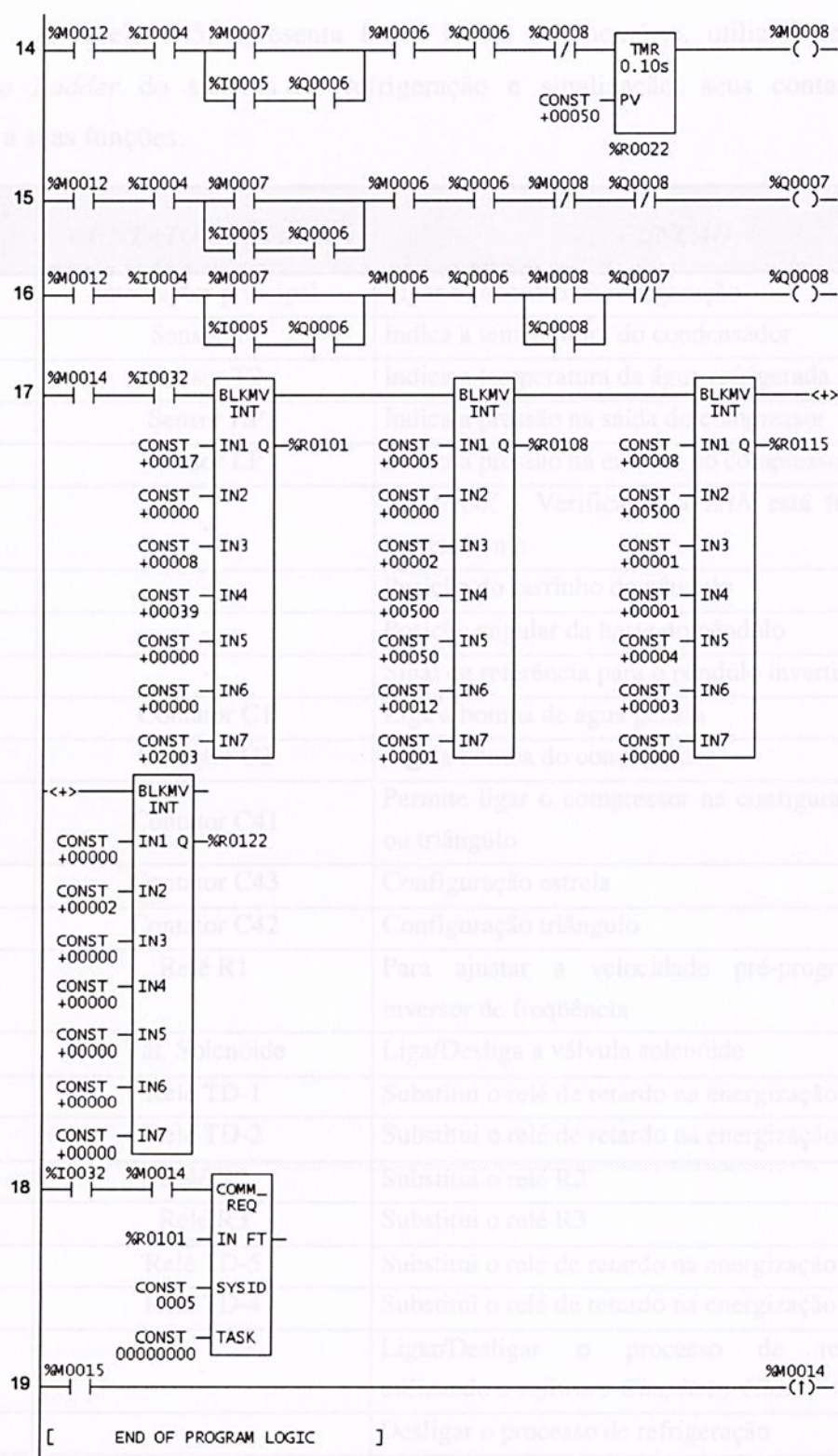


Figura B.26 - diagrama Ladder do sistema de refrigeração e sinalização via pêndulo invertido – parte 2/2.

A tabela B.5, apresenta todos locais de memória, utilizado na lógica de programação *Ladder* do sistema de refrigeração e sinalização, seus contatos elétricos referentes e a suas funções.

LOCAL DE MEMÓRIA	CONTATO ELÉTRICO	FUNÇÃO
%I01	Interruptor principal	Ligar o processo de refrigeração
%I02	Sensor T1	Indica a temperatura do condensador
%I03	Sensor T2	Indica a temperatura da água refrigerada
%I04	Sensor HP	Indica a pressão na saída do compressor
%I05	Sensor LP	Indica a pressão na entrada do compressor
%I32	-	IFLANOK - Verifica se a LAN está funcionando corretamente
%AI01	-	Posição do carrinho do pêndulo
%AI02	-	Posição angular da haste do pêndulo
%AQ01	-	Sinal de referência para o pêndulo invertido
%Q01	Contator C1	Liga a bomba de água gelada
%Q02	Contator C2	Liga a bomba do condensador
%Q06	Contator C41	Permite ligar o compressor na configuração estrela ou triângulo
%Q07	Contator C43	Configuração estrela
%Q08	Contator C42	Configuração triângulo
%Q09	Relé R1	Para ajustar a velocidade pré-programada no inversor de frequência
%Q011	Val. Solenóide	Liga/Desliga a válvula solenóide
%M01	Relé TD-1	Substitui o relé de retardo na energização TD-1
%M03	Relé TD-2	Substitui o relé de retardo na energização TD-2
%M06	Relé R2	Substitui o relé R2
%M07	Relé R3	Substitui o relé R3
%M08	Relé TD-5	Substitui o relé de retardo na energização TD-5
%M09	Relé TD-4	Substitui o relé de retardo na energização TD-4
%M10	-	Ligar/Desligar o processo de refrigeração, utilizando o <i>software Cimplicity HMI</i>
%M11	-	Desligar o processo de refrigeração
%M12	-	Liga/Desliga o processo de refrigeração
%M13	-	Desligar o processo de refrigeração

Tabela B.5 - Locais de memória utilizados na lógica de programação.



LOCAL DE MEMÓRIA	CONTATO ELÉTRICO	FUNÇÃO
%M14	-	Memória que liga a <i>COMM_REQ</i>
%M15	-	Utilizado para ligar a <i>COMM_REQ</i> remotamente, utilizando o <i>software Cimplicity HMI</i>
%R16	-	Registro de temporizador
%R19	-	Registro de temporizador
%R22	-	Registro de temporizador
%R25	-	Registro de temporizador
%R101- %R123	-	Configurar a <i>COMM_REQ</i>

Tabela B.6 - Locais de memória utilizados na lógica de programação.

Na *RUNG 1* tem-se o contato NA %I01 e em paralelo a ele o contato NA %M10. Estes dois contatos têm a mesma função: ajustar para o nível lógico “1” o *bit* de memória %M12, que será utilizado para ligar o processo controlado. A diferença entre utilizar a memória %I01 (entrada 1 do CLP1) e %M10, é que uma é feita localmente e a outra é feita de forma remota. O acesso à memória %I01 é feito através de uma chave NA ligada na entrada do CLP1, enquanto que o *bit* %M10 é controlado remotamente através do *software* de supervisão e controle *Cimplicity HMI*.

A *RUNG 2* e a *RUNG 3* são semelhantes. Sempre que o *bit* de memória %M10 passar do nível lógico “1” para “0”, energizará durante a próxima varredura o *bit* de memória %M13, que será utilizado na *RUNG 4* para desenergizar %M12, desligando o processo. O mesmo ocorre quando o *bit* da memória %I01 passa do nível “1” para “0”, porém, além de desligar o processo, quando essa memória passa para “0” ela também leva para zero o *bit* %M10.

As *RUNGS 5, 6, 7 e 8* são semelhantes ao diagrama elétrico, mudando apenas a simbologia adotada. Na *RUNG 9* tem-se a bobina de saída %Q09, usada para energizar a bobina do relé (ver figura B.24b), que atuará no borne de controle do inversor de frequência, ajustando as velocidades pré-programadas do ventilador da torre. As *RUNGS 10 à 16* são semelhantes ao diagrama elétrico, mudando-se apenas as simbologias.

Nas *RUNGS 17 e 18* é configurado o canal de comunicação entre o CLP1 (sistema de refrigeração) e o CLP2 (processo de mistura), que irá ler no CLP2 o registro %R500 e guardar no CLP1 na memória de saída %AQ01.

O registro %R500 do CLP2 guarda a informação referente à etapa de funcionamento do processo de mistura, onde: i) seu valor é 0 quando as válvulas solenóides são abertas ou quando o processo de aquecimento está sendo executado, ii) é -1600 enquanto a etapa de mistura estiver em funcionamento, e iii) é +1600 quando o processo de transferência da mistura para o depósito C estiver em andamento.

Assim o CLP1 recebe essas informações e as encaminha para o registro de saída %AQ01, que após a conversão A/D fornece à saída 1 os sinais 0, -0,5 e +0,5 Vcc, correspondentes aos valores 0, -1600 e +1600 counts, respectivamente. Esses sinais são utilizados como sinais de referência de posição para o pêndulo invertido, que equilibrará a haste buscando essas posições como referência.

A RUNG 19 tem a função de ligar o COMM_REQ e enviar as informações para os registros referentes a configuração correta da comunicação (RUNG 17).

B.3.3 Sistema Pêndulo Invertido

O pêndulo invertido é um sistema muito interessante, seja do ponto de vista do seu equacionamento, seja no estudo e enfoque de controle. Considere o sistema do pêndulo invertido como uma haste presa em uma de suas extremidades a um carrinho que move-se sobre um trilho, conforme a figura B.27.

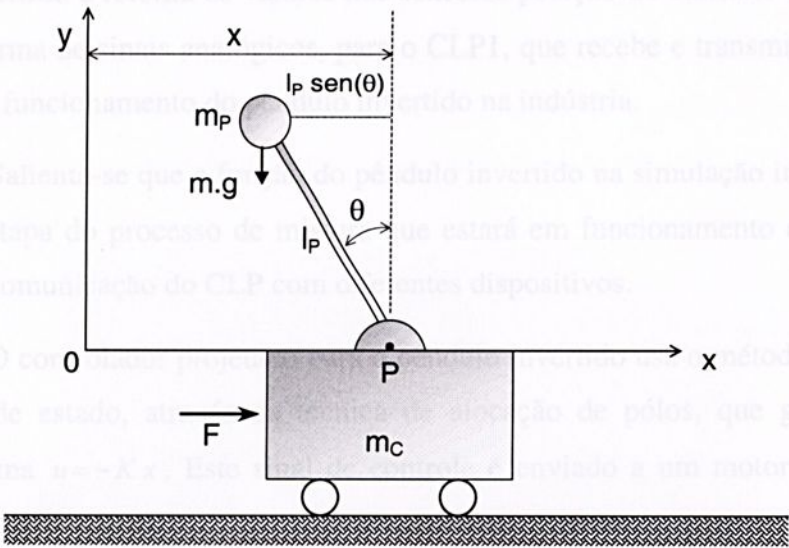


Figura B.27 - Pêndulo Invertido

Onde,

- F → somatória das forças atuantes no sistema;
- m_c → massa do carro;
- m_p → massa da bola fixada na extremidade da haste;

$l_p \rightarrow$ comprimento da haste;
 $x \rightarrow$ deslocamento do corpo de massa m_c , e
 $\theta \rightarrow$ deslocamento do corpo de massa m_p .

Como se pode observar, o modelo do pêndulo invertido é um sistema instável, dado que a haste pode cair quando é solta na posição vertical. O objetivo do sistema de controle é manter a haste em pé (90°) equilibrada sobre o carrinho que se move no trilho, mesmo com a presença de perturbações. Logo, a haste, deslocada angularmente da sua posição de equilíbrio, deverá voltar ao equilíbrio, ao mesmo tempo em que o carrinho sobre o trilho procura sua posição numa determinada referência, fornecida pelo CLP1.

Com sinais elétricos provindos do CLP1 muda-se a referência, dependendo apenas da etapa de funcionamento do processo de mistura, controlado pelo CLP2. Logo o carrinho tentará equilibrar a haste em lugares diferentes do seu trilho, dependendo apenas do sinal que o CLP1 enviará; este sinal poderá ser 0V, -0.5V ou +0.5V.

Para projetar o controlador, do pêndulo invertido, passa-se por uma etapa de modelagem e simulação que não são apresentados na monografia, sendo o objetivo, deste item, destacar a importância do CLP, dado sua capacidade de comunicação com diversos dispositivos, neste caso, com uma placa de aquisição e transmissão de dados, ligada a um PC.

Esta placa recebe as informações enviadas pelo CLP1 para realizar o controle do pêndulo invertido e retorna os valores dos sensores posição do carrinho e posição angular da haste, em forma de sinais analógicos, para o CLP1, que recebe e transmite para outro PC¹ supervisionar o funcionamento do pêndulo invertido na indústria.

Salienta-se que a função do pêndulo invertido na simulação industrial é apenas de sinalizar a etapa do processo de mistura que estará em funcionamento e de evidenciar a capacidade de comunicação do CLP com diferentes dispositivos.

O controlador projetado para o pêndulo invertido usa o método de controle por realimentação de estado, através da técnica de alocação de pólos, que gera um sinal de controle da forma $u = -Kx$. Este sinal de controle é enviado a um motor *dc*, acoplado ao carrinho, que terá a função de equilibrar a haste e de procurar a posição de referência no trilho.

¹ Computador ligado à rede, no qual é instalado o *software Cimplicity HMI*.



A figura B.28 mostra o modelo do pêndulo invertido, disponível no laboratório de controle da FEIS, adquirido junto à empresa canadense *Quanser Consulting*. O modelo é constituído por uma haste articulada sobre o ponto central do carrinho e o mesmo fica sobre um trilho, onde pode se mover em dois sentidos, dependendo do sinal enviado ao motor *dc*, embutido no carrinho. Esta haste mover-se apenas num sistema bidimensional, ou seja, sobre o ponto de sua base articulada no carrinho.

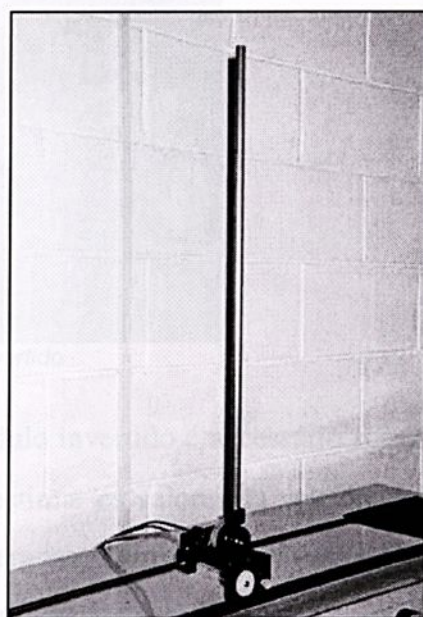
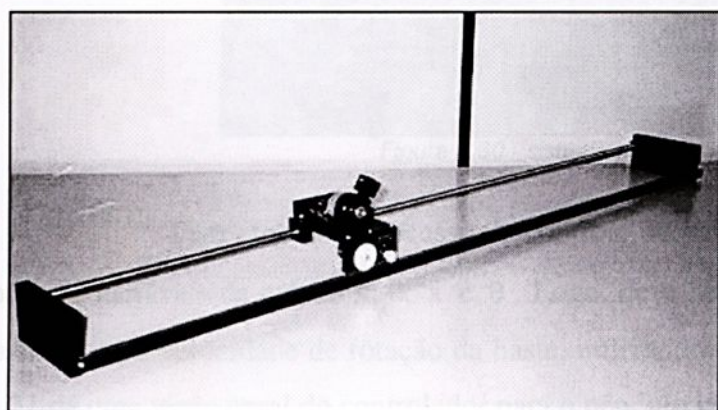


Figura B.28 - carrinho e haste do modelo de pêndulo invertido

Além do motor *dc*, este carrinho é constituído também por dois sensores que tem a função de registrar a posição do carro sobre o trilho e a posição angular da haste em relação a uma linha vertical, ou seja, em relação à posição de equilíbrio para o pêndulo (90° em relação ao trilho, quando o trilho estiver sobre uma região plana).

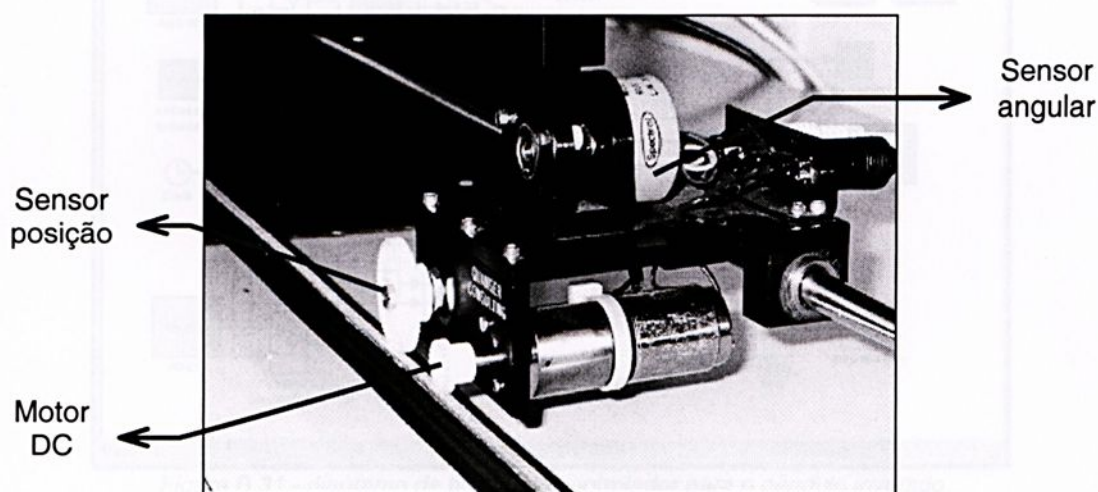


Figura B.29 - sensores e motor DC embutidos no carrinho

A figura B.30 dá uma visão geral do sistema pêndulo invertido, bem como todas as suas ligações.

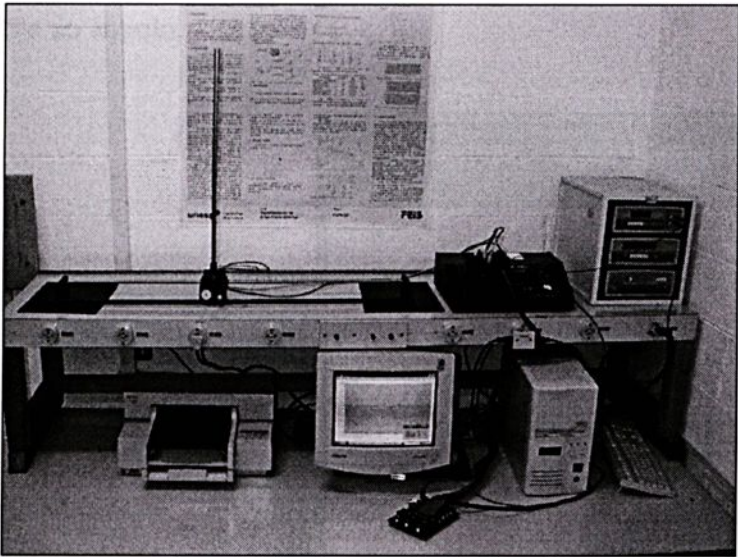


Figura B.30 - sistema pêndulo invertido

Para efetuar o processo de controle do pêndulo invertido é necessário acesso a todas as variáveis de estado x , θ , $\dot{x}$ e $\dot{\theta}$. Logo, deve-se estimar os valores da velocidade do carrinho e da velocidade de rotação da haste, utilizando para isso filtros derivativos. A figura B.31 dá uma visão geral do controlador para o pêndulo invertido.

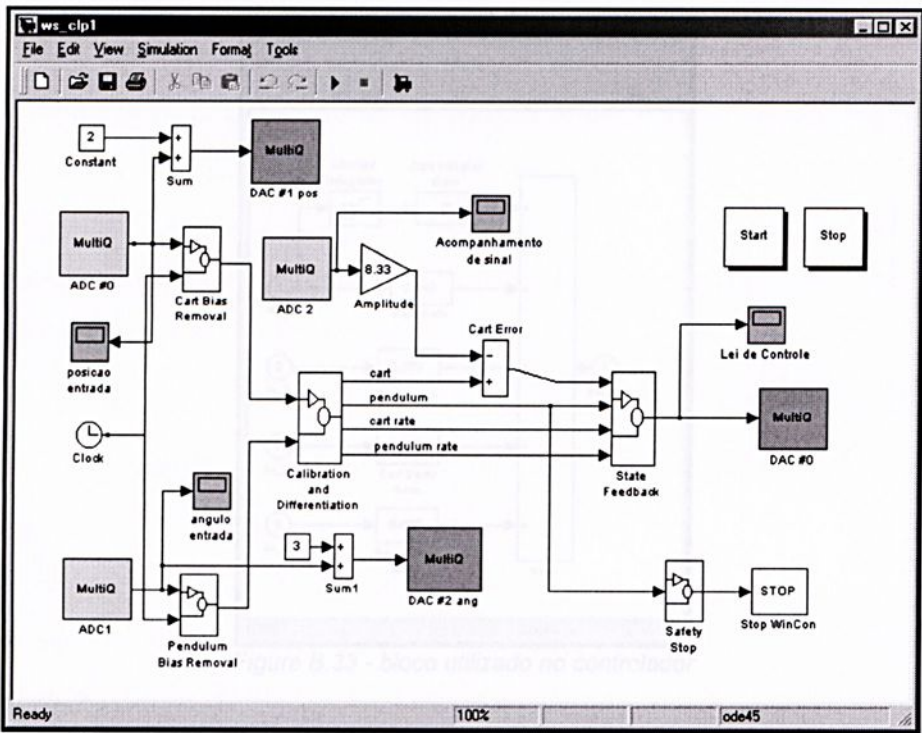


Figura B.31 - diagrama de blocos do controlador para o pêndulo invertido

As figura B.32 e B.33 são figuras que mostram os blocos de controle expandidos, detalhando os blocos necessários para o funcionamento correto do controlador. Por exemplo, a figura B.32b mostra os filtros derivativos para obtenção das velocidades referentes à posição e ao ângulo.

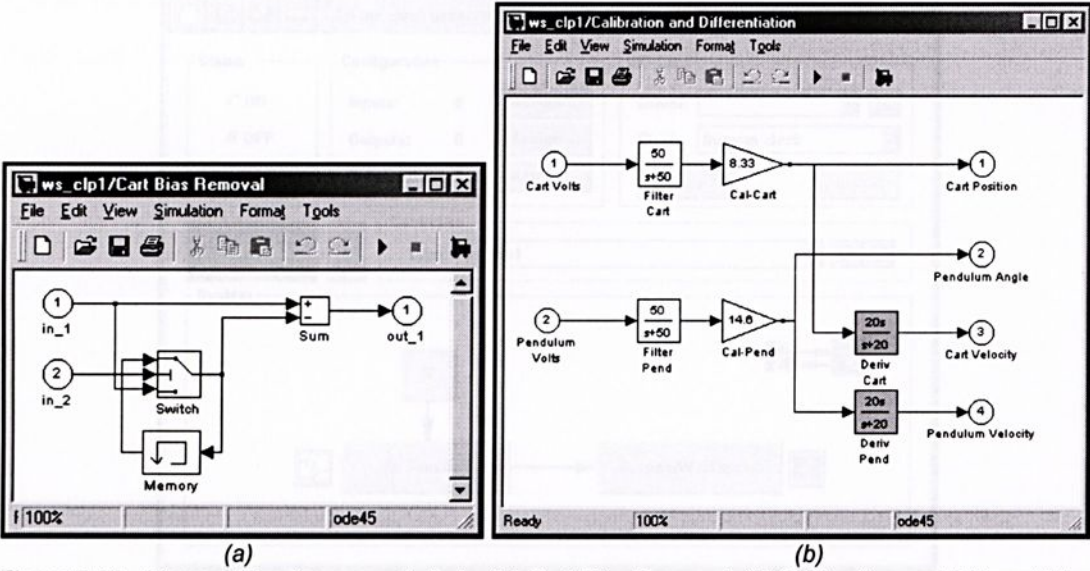


Figura B.32 - blocos utilizados no controlador (a) Cart Bias Removal, (b) Calibration and Differentiation

A figura B.33 mostra os ganhos para cada estado que, após multiplicados e somados, resulta na lei de controle do tipo $u = -Kx$.

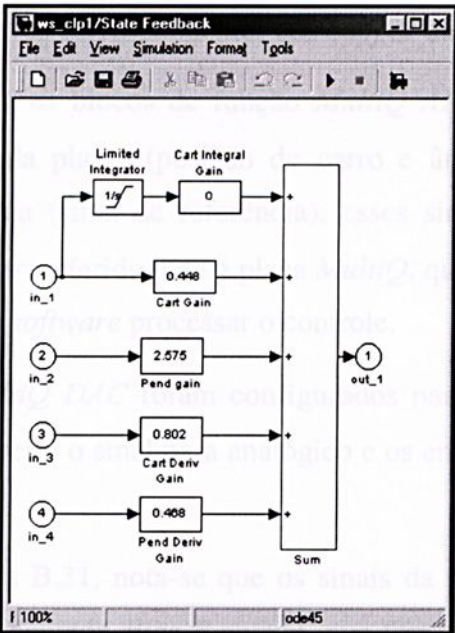


Figura B.33 - bloco utilizado no controlador

O processamento do sinal de controle é feito no *software Matlab with Simulink 4.2c*, junto com um *toolbox* fornecido pela *Quanser Consulting* chamado *Wincon V2.0*, conforme mostrado na figura B.34 a seguir.

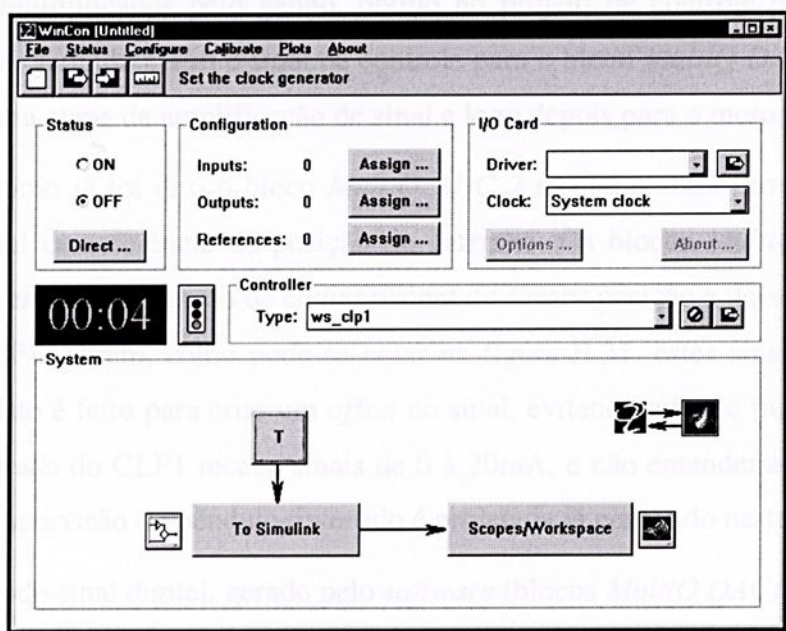


Figura B.34 - janela do programa Wincon 2.0.

Na figura B.34, o bloco *To Simulink* direciona para o controle em diagrama de blocos da figura B.31, onde se pode alterar os ganhos de realimentação em tempo real, melhorando o desempenho do controlador. O bloco *Scopes/Workspace* abre a tela gráfica onde se pode visualizar as curvas correspondentes aos *Scopes* da figura B.31.

Na figura B.31 os blocos de função *MultiQ ADC* foram configurados para receberem os sinais vindos da planta (posição do carro e ângulo da haste) e o sinal de referência que o CLP informa (sinal de referência); esses sinais são recebidos em forma analógica na placa terminal e transferido para a placa *MultiQ*, que converte o sinal para digital e envia as informações para o *software* processar o controle.

Os blocos *MultiQ DAC* foram configurados para transferir os dados digitais para a placa *MultiQ*, que converte o sinal para analógico e os envia para a placa terminal, que está ligada ao CLP1.

Ainda na figura B.31, nota-se que os sinais da posição do carro e ângulo da haste passam por um bloco chamado *Bias Removal*, que tem a função de gerar o ponto de referência quando o bloco *MultiQ ADC 2* envia sinal zero. Saindo deste bloco o sinal passa pelo bloco *Calibration and Differentiation*, que inicialmente filtra o sinal, eliminando os



ruídos, depois converte o sinal de Volts para a unidade utilizada no projeto do controlador e finalmente estima, através dos filtros derivativos, as variáveis $\dot{x}$ e $\dot{\theta}$.

Por último os quatro estados, dois com acesso direto e os outros dois estimados, são multiplicados pelo ganho, obtido no projeto de controle por realocação de pólos, e somados, gerando assim o sinal de controle para o bloco *MultiQ DAC#0*, que enviará este sinal para uma etapa de amplificação de sinal e logo depois para o motor *dc* do carrinho.

Como já foi dito o bloco *MultiQ ADC 2* recebe o sinal fornecido pelo CLP1 para gerar o sinal de referência da posição do carrinho. Os blocos *MultiQ DAC #1 pos* e *MultiQ DAC #2 ang*, tem a função de enviar o sinal do sensor posição e do sensor ângulo para a entrada do CLP1, porém, como pode-se notar na figura B.31, estes sinais são somados a uma constante. Isto é feito para criar um *offset* no sinal, evitando que ele fique negativo, pois o módulo de entrada do CLP1 recebe sinais de 0 à 20mA, e não entenderia o sinal negativo. Logo, a tela de supervisão do pêndulo invertido é projetada já pensando neste *offset* do sinal.

Todo sinal digital, gerado pelo *software* (blocos *MultiQ DAC*), é enviado para a placa *MULTIQ*, que converte o sinal em analógico o envia à placa terminal, externa a torre do *PC*, conforme esboço mostrado na figura B.35.

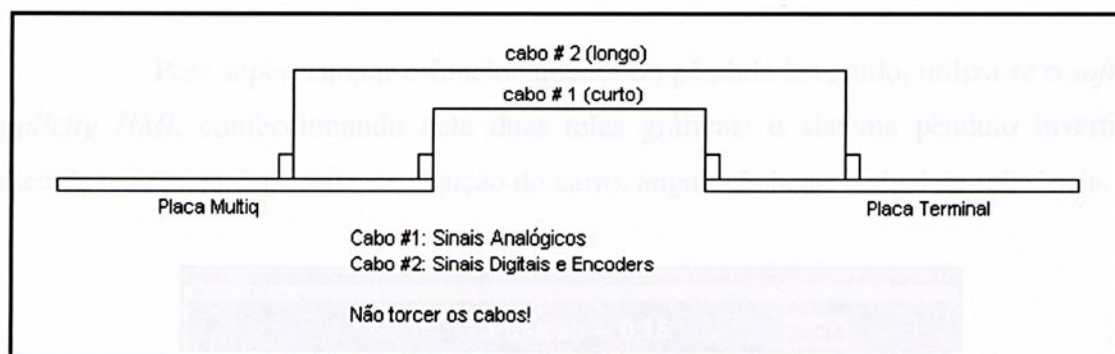


Figura B.35 - Interligação das placa *MULTIQ™* e *TERMINAL*.

A placa terminal envia o sinal analógico, provindo do bloco *MultiQ DAC #0*, para um amplificador de potência, apresentado na figura B.36, que aumenta a potência do sinal, antes de alimentar o motor *dc*.

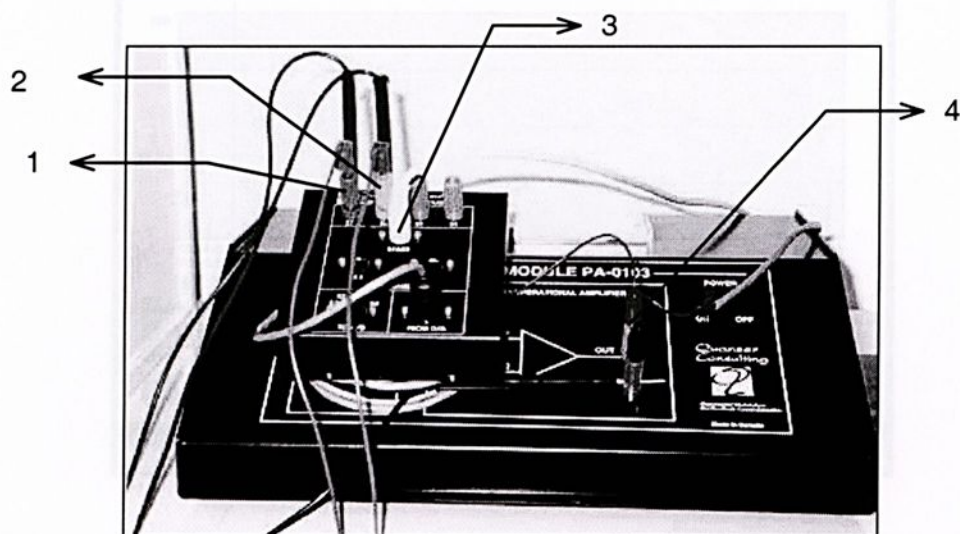


Figura B.36 - módulo de potência, utilizado para amplificar o sinal DC para o motor do carrinho e local onde temos o acesso as variáveis x e θ .

Onde,

- (1) sinal do sensor posição que será enviado ao sistema de controle;
- (2) sinal do sensor ângulo que será enviado ao sistema de controle;
- (3) sinal dos sensores ângulo e posição, enviado do carrinho e da haste para o módulo de potência, e
- (4) sinal amplificado de controle enviado ao motor DC do carrinho.

Para supervisionar o funcionamento do pêndulo invertido, utiliza-se o *software Cimplicity HMI*, confeccionando nele duas telas gráficas: o sistema pêndulo invertido e gráfico de tendência das variáveis posição do carro, ângulo da haste e sinal de referência.

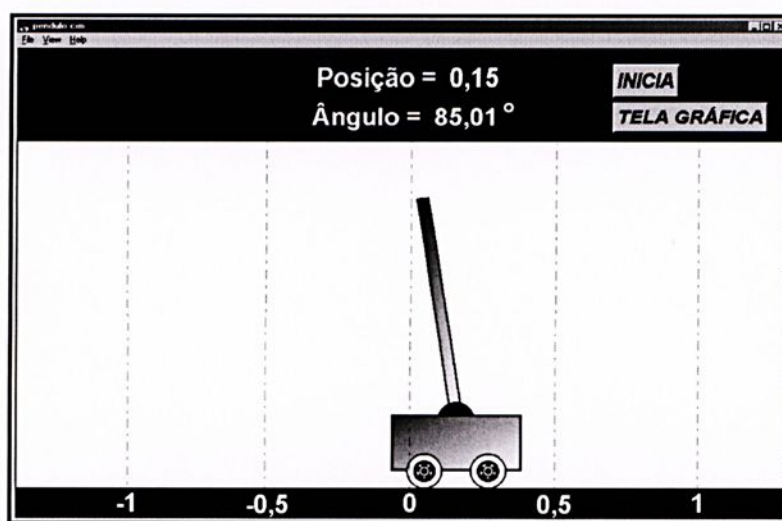


Figura B.37 - sistema pêndulo invertido.

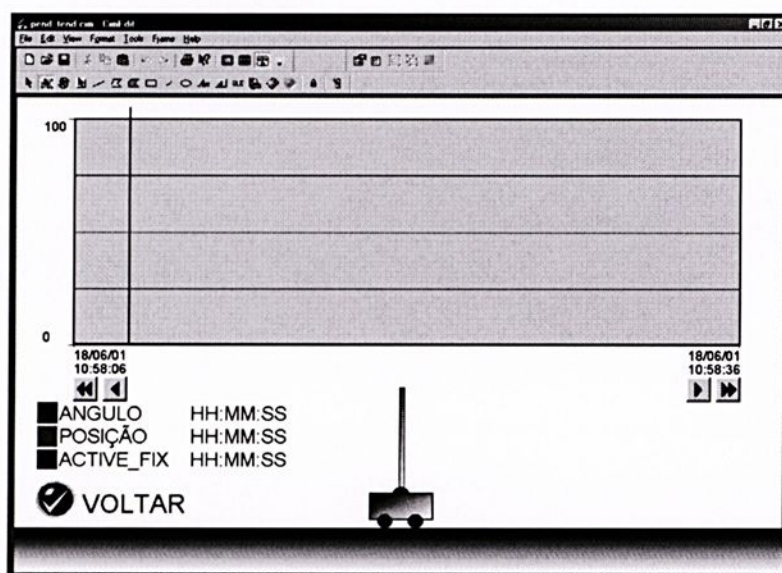


Figura B.38 - gráfico de tendência.

Através destas telas pode-se então verificar qual a etapa do processo de mistura está sendo realizada, observando apenas o local de referência que o pêndulo invertido está buscando.

No apêndice C são apresentadas a confecção e configuração dos pontos, na tela de supervisão e controle, apenas do processo de mistura, visto que é um exemplo bastante amplo, com o qual já se pode verificar as possibilidades e capacidades do *software*.

CRIAÇÃO DAS TELAS DE SUPERVISÃO

Após implementação da lógica de controle nos CLPs, utilizando o *software Logixmaster TCP/IP Vx.62*, inicia-se o estado do *software* de supervisão e controle *Cimplicity HMI*, onde o objetivo é criar telas gráficas que apresentem o funcionamento geral da proposta final que é a simulação de uma indústria de pequeno porte.

A criação de telas de supervisão envolve muitos detalhes e esses serão explicados no decorrer da apresentação deste apêndice. Este apêndice apresenta um exemplo prático por ser a forma mais rápida e fácil de ensinar a programação e supervisão. Por conveniência, escolheu-se o processo de destilação e aquecimento de líquidos, como exemplo a ser seguido. As telas das demais aplicações industriais utilizadas a mesma metodologia, por

Apêndice C

Software de supervisão e controle: Cimplicity HMI

Uma informação inicial, sobre o *software*, é muito importante, dando uma visão geral dos problemas que se pode encontrar no projeto de supervisão e controle:

O *software Cimplicity HMI* é um *software SCADA* (*Supervisory Control and Data Acquisition*) projetado para que usuários possam monitorar e controlar a produção através de comunicação com outros dispositivos, neste caso em particular, com os CLPs. As informações podem ser acessadas localmente, utilizando uma grande rede de dispositivos, onde o *PC*, com o *software* de supervisão, pode estar em um ponto desta rede, ou remotamente, através da *Web*, *modem* e até mesmo em comunicação *wireless*, por exemplo, *um bip*.

Para iniciar um novo projeto deve-se primeiro abrir o *software SIMPLICITY HMI Workbench*, que é a tela principal de todo o projeto no *Cimplicity*, onde todos os dados referentes ao projeto estarão localizados; fazendo uma analogia, o *Workbench* seria como o *Windows Explorer*, tratando das informações contidas num computador pessoal.



CRIAÇÃO DAS TELAS DE SUPERVISÃO

Após implementação da lógica de controle nos CLPs, utilizando o *software Logicmaster TCP/IP V9.02*, inicia-se o estudo do *software* de supervisão e controle *Cimplicity HMI*, onde o objetivo é criar telas gráficas que apresentem o funcionamento geral da proposta final que é a simulação de uma indústria de pequeno porte.

A criação de telas de supervisão envolve muitos detalhes e esses serão explicados no decorrer da apresentação deste apêndice. Este apêndice apresenta um exemplo prático por ser a forma mais rápida e fácil de ensinar a programação e supervisão. Por conveniência, escolheu-se o processo de mistura e aquecimento de líquidos, como exemplo a ser seguido. As telas dos demais processos são confeccionadas utilizando a mesma metodologia, por isso neste item apresenta-se apenas a confecção de uma tela.

Uma informação inicial, sobre o *software*, é muito importante, dando uma visão geral dos problemas que se pode encontrar no projeto de supervisão e controle.

O *software Cimplicity HMI* é um *software SCADA (Supervisory Control and Data Acquisition)* projetado para que usuários possam monitorar e controlar a produção através de comunicação com outros dispositivos, neste caso em particular, com os CLPs. As informações podem ser acessadas localmente, utilizando uma grande rede de dispositivos, onde o *PC*, com o *software* de supervisão, pode estar em um ponto desta rede, ou remotamente, através da *Web*, *modems* e até mesmo em comunicação *wireless*, por exemplo, um *bip*.

Para iniciar um novo projeto deve-se primeiro abrir o *software CIMPLICITY HMI Workbench*, que é a tela principal de todo o projeto no *Cimplicity*, onde todos os dados referentes ao projeto estarão localizados; fazendo uma analogia, o *Workbench* seria como o *Windows Explorer*, tratando das informações contidas num computador pessoal.



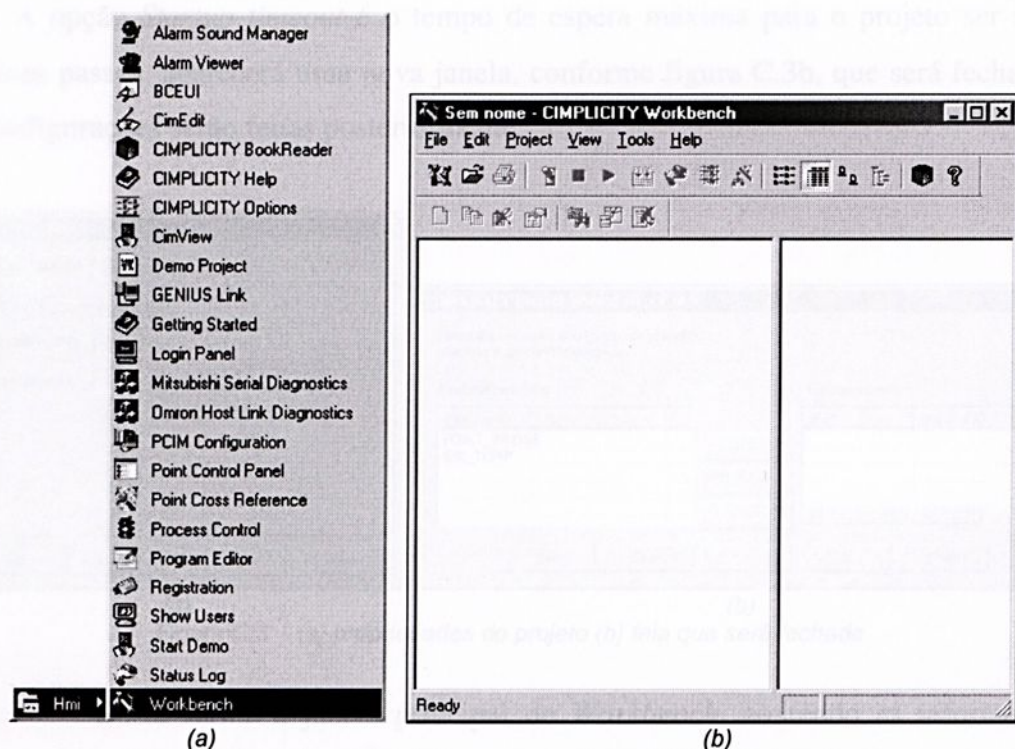


Figura C.1 - (a) ícone de atalho para o software, (b) janela do software.

Executando o comando *File* → *New* → *Project*, será aberto uma nova janela onde deve-se informar o nome do projeto, as opções utilizadas no projeto e o protocolo de comunicação. Para o caso explicado neste item dá-se o nome do projeto de *INDUSTRIA*, disponibilizando todos recursos do *software* (embora o laboratório da FEIS não tenha a licença de alguns recursos, que impossibilita serem usados) e escolhe-se o protocolo de comunicação *Ethernet* para CLPs série 90. Esses passos são descritos na figura C.2.

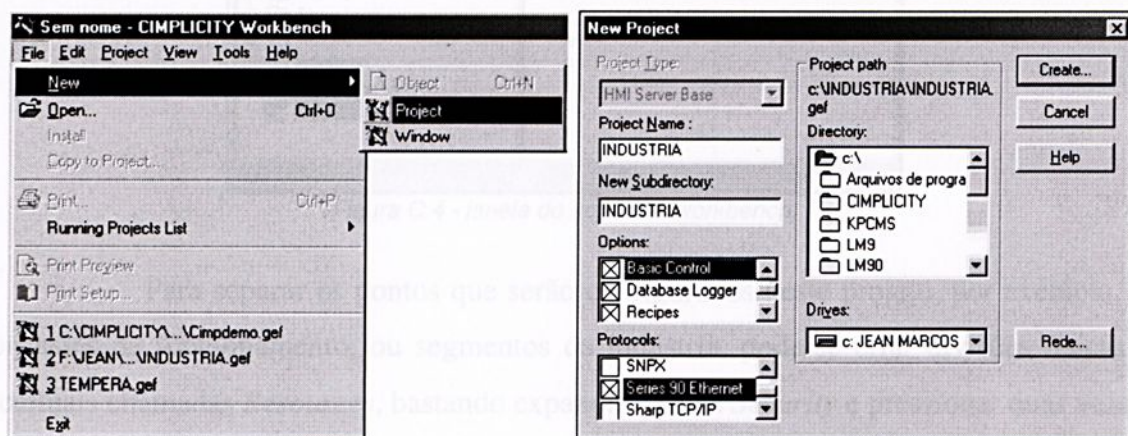


Figura C.2 - criando um novo projeto.

Pressionando o botão *Create* abrirá a próxima janela, onde deve-se selecionar a opção *Enable project broadcast*, disponibilizando o projeto na rede para ser lido por outros

pontos. A opção *Startup timeout* é o tempo de espera máxima para o projeto ser iniciado. Após estes passos, aparecerá uma nova janela, conforme figura C.3b, que será fechada, pois essas configurações serão feitas posteriormente.

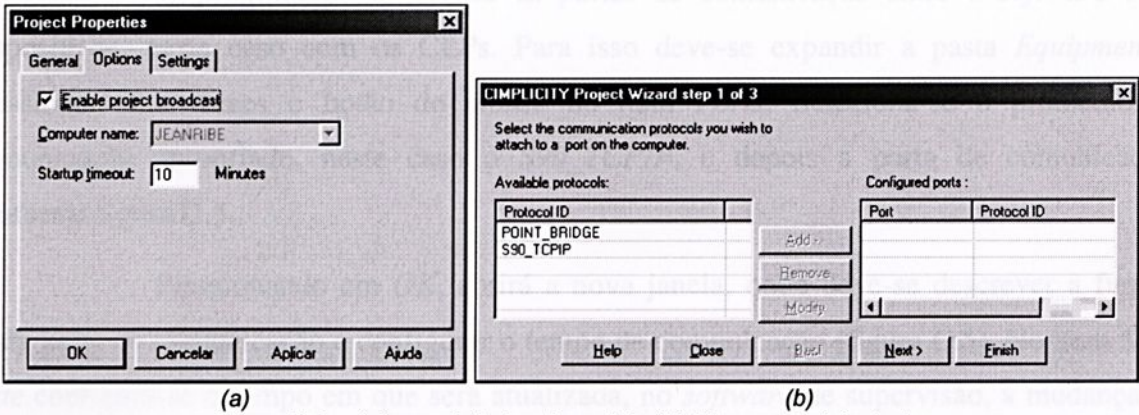


Figura C.3 - (a) propriedades do projeto (b) tela que será fechada.

Desta forma a janela principal do *Workbench*, contendo as informações da configuração do projeto, é criada. Os itens do *menu* são facilmente compreensíveis, bastando então explicar as funções que aparecem no plano esquerdo da janela, semelhante à estrutura em árvore do *Windows Explorer*.

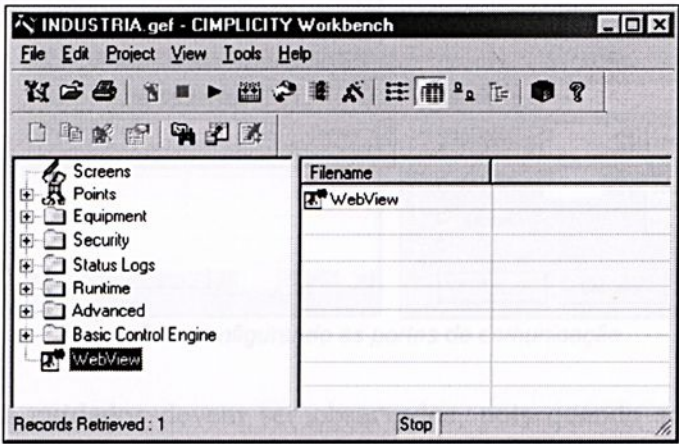


Figura C.4 - janela do software Workbench.

Para separar os pontos que serão configurados neste projeto, por exemplo, em prioridades de funcionamento, ou segmentos da indústria, pode-se criar divisões físicas ou conceituais chamadas *Resources*, bastando expandir a pasta *Security* e pressionar duas vezes o botão esquerdo do *mouse* sobre o item *Resources*, escolhendo um nome para o segmento. Isto é útil na organização do projeto e também quando se necessita criar rotinas de alarmes em que alguns pontos são mais importantes que outros.

Contudo, neste exemplo, não são criados novos *Resources*, visto que o próprio *software* já vem com três *Resources* configurados e não serão priorizados pontos de alarme neste projeto.

O próximo passo é criar as portas de comunicação entre o *software* e os dispositivos, neste caso com os CLPs. Para isso deve-se expandir a pasta *Equipment* e pressionar duas vezes o botão do *mouse* no item *Ports*. Seleciona-se o protocolo de comunicação apropriado, neste caso o *S90_TCPIP*, e depois a porta de comunicação, conforme figura C.5.

Pressionando em *OK*, abrirá a nova janela, onde deve-se descrever a função desta porta de comunicação e configurar o tempo de comunicação (figura C.5). No item *Scan Rate* configura-se o tempo em que será atualizada, no *software* de supervisão, a mudança de um estado de memória do dispositivo CLP. No item *Retry Count* escolhe-se a quantidade de vezes que o *software* tentará buscar os dados caso não consiga na varredura especificada por *Scan Rate*¹.

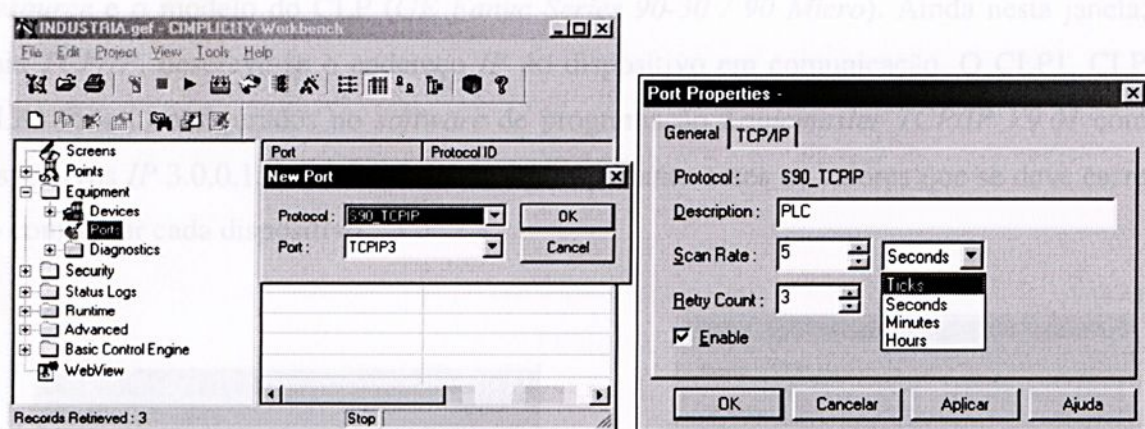


Figura C.5 - configurando as portas de comunicação.

Alguns cuidados devem ser observados, pois quando se transmite dados de forma rápida, pode-se congestionar a rede causando conflito ou perda de informações, por isso, em projetos com muitos pontos não é viável colocar um tempo muito baixo de atualização de dados. No projeto da simulação industrial foi utilizado o tempo mais baixo, ou seja, 100 milissegundos, por se tratar de um projeto com poucos pontos em rede.

Na guia *TCP/IP*, mantém-se a configuração padrão para os *Timeout*. Após configurar as portas de comunicação aparece a tela conforme a figura C.6.

¹ Quanto menor o valor do *Scan Rate* mais rápida será a apresentação do dado no *software Cimplicity HMI*, chegando muito próximo do tempo real quando colocamos na máxima velocidade (100ms ou 1 Ticks).

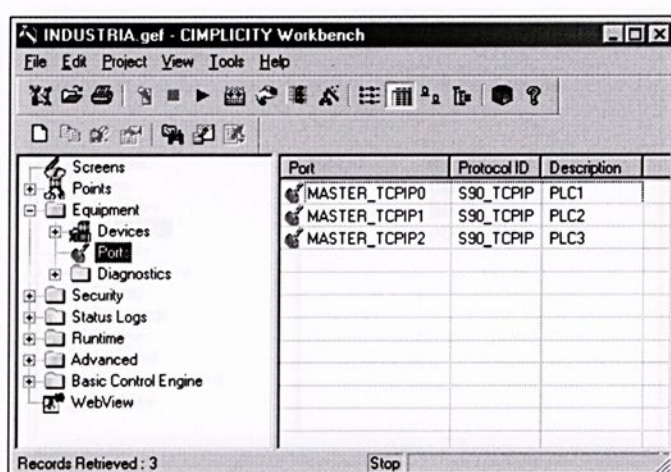


Figura C.6 - janela após configuração de três portas de comunicação.

O próximo passo é configurar os dispositivos que estarão em comunicação com o *software*, neste caso o CLP1, CLP2 e o CLP3. Para isso, ainda dentro na pasta *Equipment*, pressiona-se duas vezes no botão esquerdo do *mouse* sobre o item *Devices*, abrindo uma janela onde se nomeia o dispositivo e escolhe-se a porta de comunicação.

Após pressionar em *OK*, uma nova janela se abrirá, onde deve-se escolher o *Resource* e o modelo do CLP (*GE Fanuc Series 90-30 / 90 Micro*). Ainda nesta janela, na guia *TCP/IP*, descreve-se o endereço *IP* do dispositivo em comunicação. O CLP1, CLP2 e CLP3 foram configurados no *software* de programação *Logicmaster TCP/IP V9.01* com os respectivos *IP* 3.0.0.1, 3.0.0.2 e 3.0.0.3, sendo portanto esses os valores que se deve escrever ao configurar cada dispositivo.

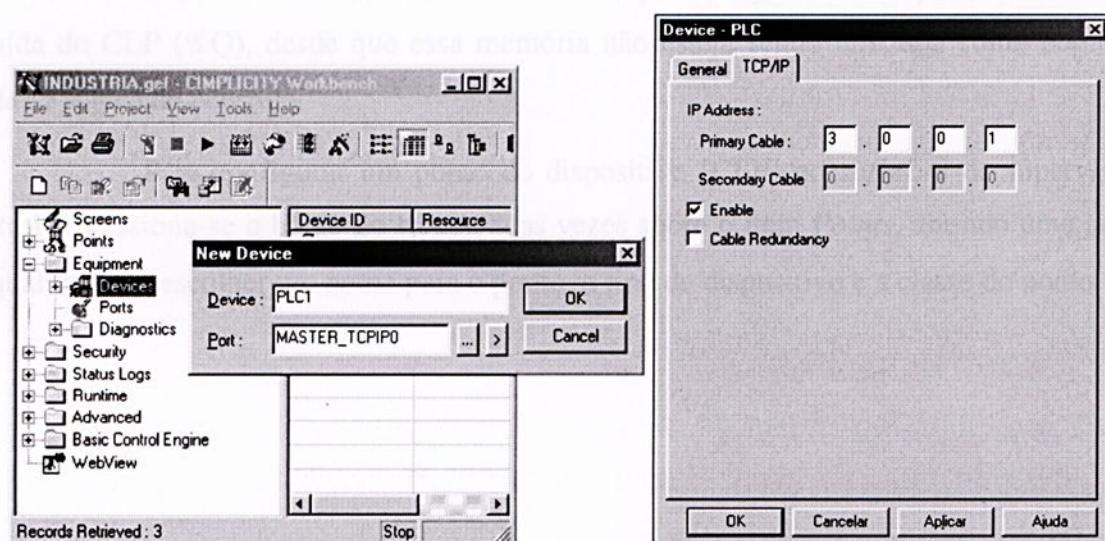


Figura C.7 - configuração dos dispositivos (CLPs).

Após configurar todos os três CLPs, resultará na tela, conforme a figura C.8, lembrando que o *Resource* escolhido poderia ser outro.

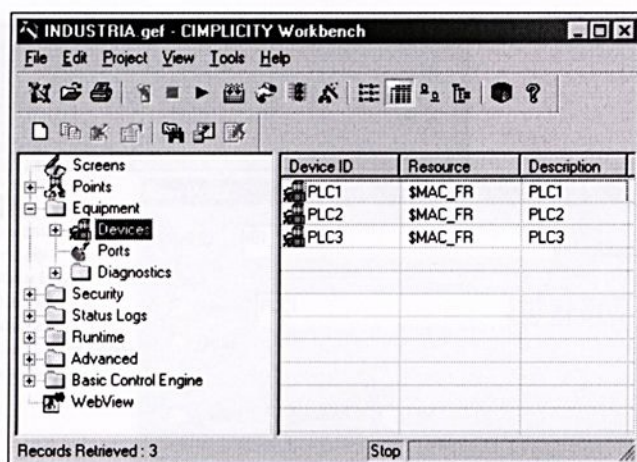


Figura C.8 - janela após a configuração dos dispositivos.

Já configurados os dispositivos e as portas de comunicação pode-se inserir os pontos dos dispositivos com os quais o *software* irá fazer comunicação, seja ela para ler um dado da memória ou para escrever nele.

Um fato muito importante a ser mencionado é que o *software* de supervisão e controle não escreverá sobre um dado de memória que esteja sendo utilizado como bobina de saída do CLP. Por exemplo, o dado de memória %Q01 do CLP2 (processo de mistura) é utilizado na lógica, escrita pelo *software Logimaster TCP/IP*, para abrir ou fechar as válvulas solenóides, logo não se pode mudar o valor de %Q01 através do *Cimplicity*, mas pode-se criar um intertravamento com um dado de memória, em série com a bobina, para permitir ou não que as válvulas sejam abertas.

Pode-se, é claro, mandar uma informação do *software* de supervisão direto para a saída do CLP (%Q), desde que essa memória não esteja sendo utilizada como bobina de saída na lógica de controle.

Para configurar um ponto do dispositivo (CLP) no *software* de supervisão e controle, pressiona-se o botão do *mouse* duas vezes sobre o item *Points*, abrindo uma janela, na qual deve-se escolher um nome para o ponto, o tipo de dispositivo e a classe do ponto.

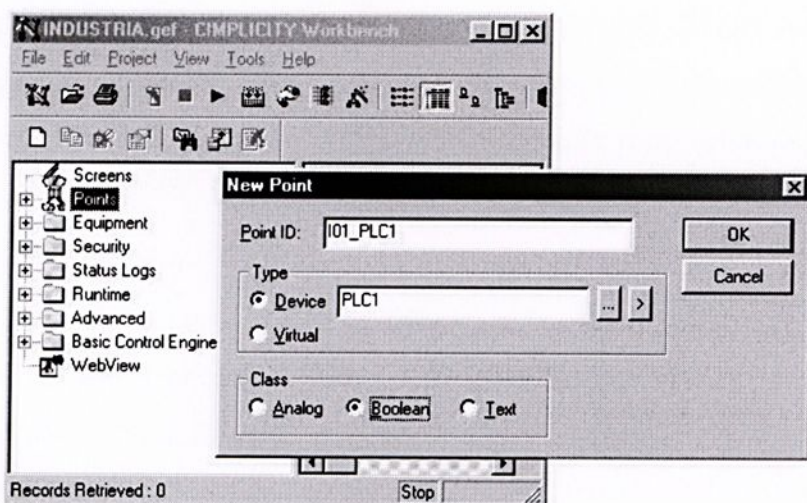


Figura C.9 - inserindo o ponto digital referente ao local de memória %I01 do CLP1.

Na figura C.9 é apresentado a configuração do ponto de memória %I01 do CLP1. Primeiro escolhe-se o nome para o ponto no campo *Point ID*, depois seleciona-se o dispositivo no item *Device* e finalmente classifica-se o ponto como digital (*Boolean*), já que a memória %I01 só pode ser “0” ou “1”. Pressionando *OK*, abrirá a janela sobre as propriedades do ponto (figura C.10), onde se insere a descrição do ponto no campo *Description* da guia *General*, e no campo *Address* da guia *Device* é inserido o endereço de memória referente a este ponto, ou seja, %I01.

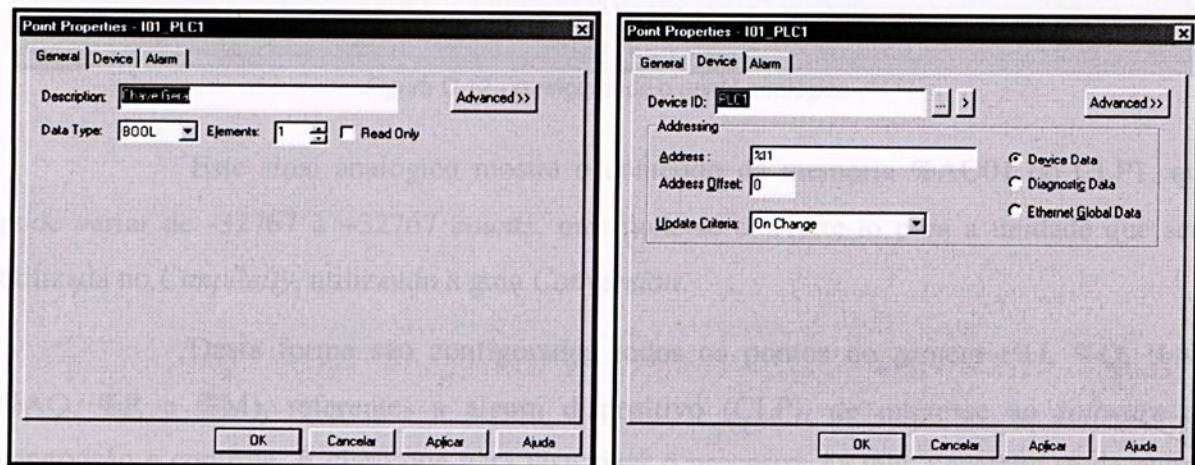


Figura C.10 - configurando as propriedades do ponto.

O *Resource* pode ser mudado escolhendo o item *Advanced* na guia *General* (figura C.10). Todos os pontos digitais dos CLPs (%I, %Q ...) são configurados desta forma. A seguir é apresentado a configuração de um ponto analógico do CLP1; neste exemplo será configurado a memória de saída analógica %AQ01, que é usada como sinal de referência para o pêndulo invertido. Os passos são semelhantes ao anterior, contudo ao invés de escolher a classe *Boolean*, deve-se escolher *Analog*, conforme as figuras C.11 e C.12.

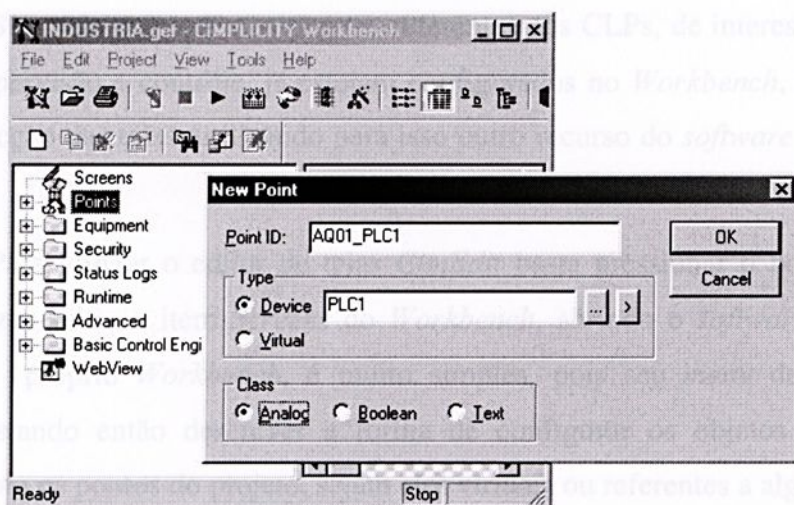


Figura C.11 - inserindo o ponto analógico referente ao local de memória %AQ01.

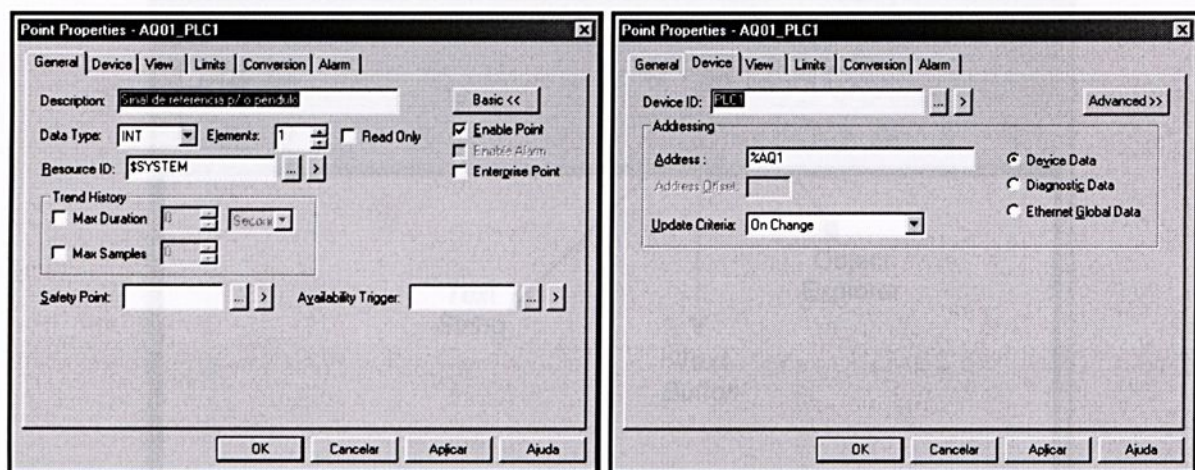


Figura C.12 - configurando o ponto analógico.

Este sinal analógico mostra o conteúdo da memória %AQ01 do CLP1, que pode variar de -32767 à +32767 *counts*, mas pode-se convertê-lo para a unidade que será utilizada no *Cimplicity*, utilizando a guia *Conversion*.

Desta forma são configurados todos os pontos do projeto (%I, %Q, %AI, %AQ, %R e %M), referentes a algum dispositivo (CLP), de interesse ao *software* de supervisão e controle. É claro que para visualizar o processo, na tela, necessita-se de outros pontos, que não necessariamente são os pontos referentes a dispositivos.

Esses pontos são chamados pontos virtuais e podem ser utilizados para criar alguma animação na tela, para ligar algum evento ou enviar informação para o CLP, para intertravar pontos fora da lógica armazenada no CLP, etc. A configuração dos pontos virtuais será mostrada no momento em que for apresentado algum recurso de animação, ou outro evento que seja necessário ponto virtual.

Supondo que todos os pontos, referentes aos CLPs, de interesse para a criação das telas de supervisão e controle, já estejam configurados no *Workbench*, pode-se iniciar a criação e confecção das telas, utilizando para isso outro recurso do *software Cimplicity HMI*: o *CimEdit*.

Para utilizar o editor de telas *CimEdit* basta pressionar o botão esquerdo do *mouse* duas vezes sobre o item *Screens* do *Workbench*, abrindo o *software* de edição que, assim como o próprio *Workbench*, é muito simples, pois seu *menu* de edição é auto-explicativo, bastando então descrever a forma de configurar os objetos que estarão em comunicação com os pontos do projeto, sejam eles virtuais ou referentes a algum dispositivo.

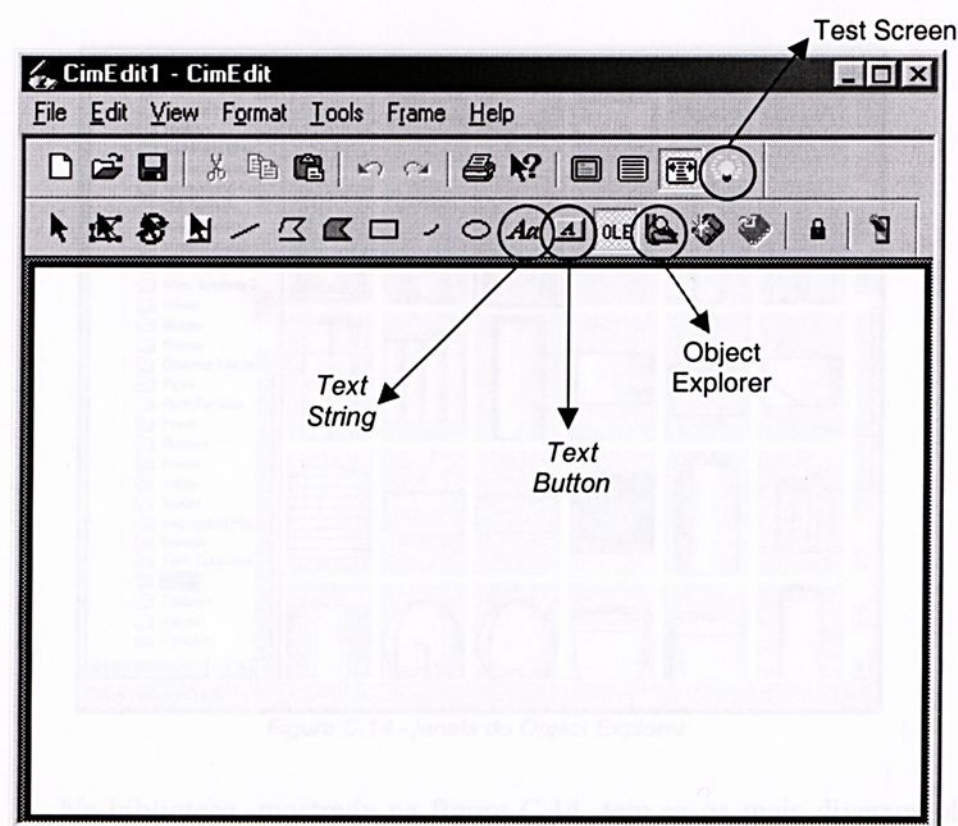


Figura C.13 - janela de edição, das telas de supervisão e controle, *CimEdit*.

Sugere-se iniciar a confecção criando as telas individuais, antes de criar a tela geral que mostra todo o processo. Por exemplo, na proposta final, a simulação industrial tem os segmentos da mistura, controle de nível e sistema de refrigeração, logo cria-se primeiro uma tela para cada processo.

C.1 TELA DE SUPERVISÃO E CONTROLE PARA O PROCESSO DE MISTURA E AQUECIMENTO DE LÍQUIDO

C.1.1 Inserindo Objetos na Tela

O primeiro passo é encontrar os objetos que poderão ser utilizados para a criação desta tela. Para isso, deve-se abrir e navegar pela biblioteca de objetos disponíveis no *software*, pressionando com o *mouse* sobre o ícone de *menu* identificado como *Object Explorer*, mostrado em destaque na figura C.13. Abrirá uma nova janela, contendo vários objetos que podem ser inseridos na tela de supervisão, conforme mostra a figura C.14.

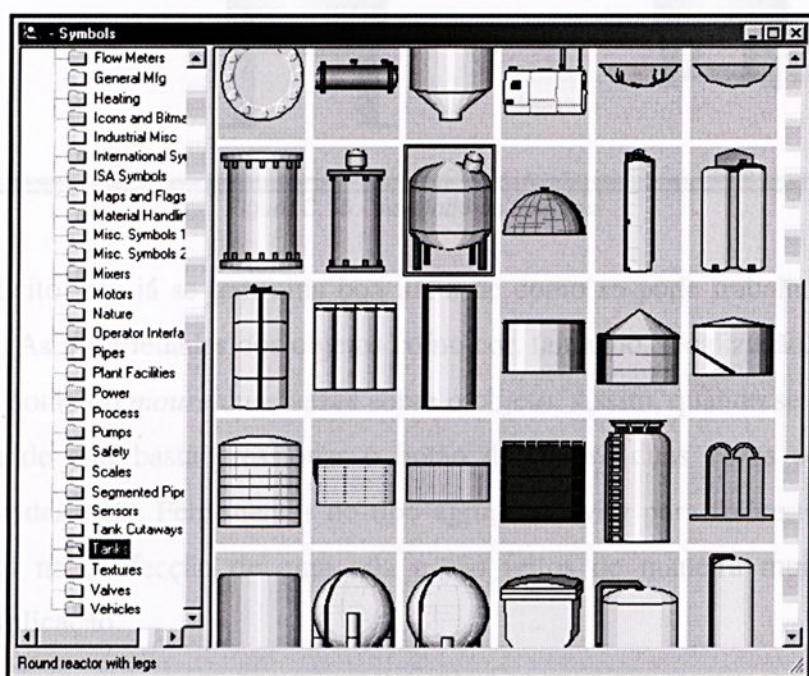


Figura C.14 - janela do Object Explorer.

Na biblioteca, mostrada na figura C.14, tem-se os mais diversos objetos que poderão ser utilizados para confeccionar a tela de controle e supervisão, entre eles os dutos, as bombas e motores, as chaves e sensores, os tanques, enfim uma gama enorme de objetos. Neste apêndice será mostrada, em detalhes, a confecção de uma tela.

Inicia-se a confecção da tela do misturador escolhendo, na biblioteca de objetos, os tanques que serão utilizados, e os posicionamos nos locais desejados. Observa-se que os dois tanques menores (depósitos A e B), da figura C.15, são os mesmos, porém em posição espelho, recurso este semelhante aos *softwares* de edição de imagens, dado o alto poder de edição que o *CimEdit* possui.

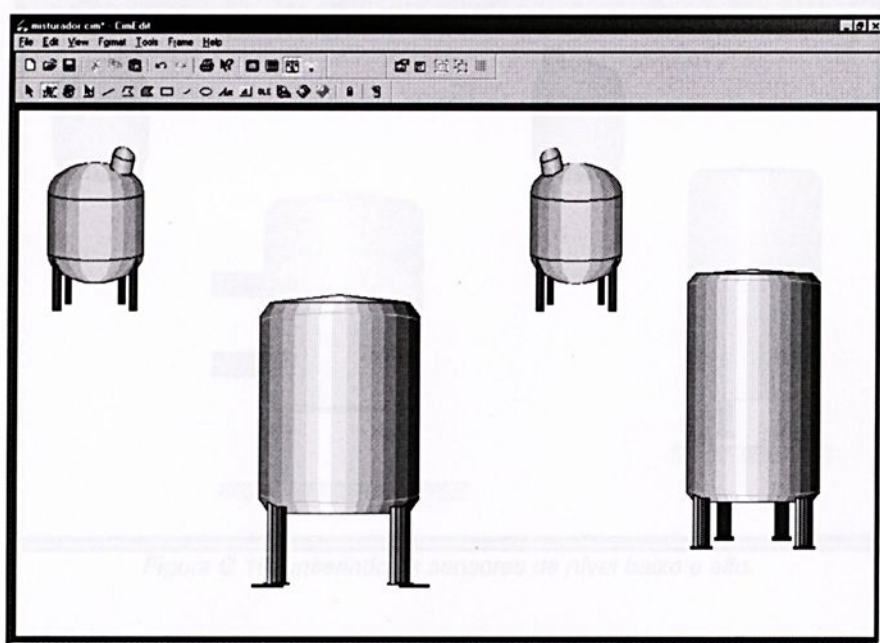


Figura C.15 - inserindo os tanques.

Feito isso já se tem uma boa idéia de como se pode trabalhar com o espaço restante da tela. As propriedades dos objetos como cor, tamanho, localização, etc, são obtidas pressionando o botão do *mouse* duas vezes sobre o objeto. Assim, quando se cria uma caixa e deseja muda-la de cor, basta pressionar o botão do *mouse* duas vezes sobre o objeto e selecionar a cor desejada. Ferramentas do tipo agrupar, enviar para frente, etc, também são muito utilizadas na confecção de uma tela e são feitos de maneira muito simples, não necessitando explicação.

Primeiro cria-se, de forma muito simples, os sensores FSH e FSL, através de caixas, cinza e verde, conforme figura C.16. Para escrever dentro da caixa verde utiliza-se a ferramenta *Text String*, mostrada na figura C.13. As caixas verde e cinza e o texto são agrupados formando um único objeto. Depois, através da biblioteca, na opção *Tanks Cutaways*, escolhe-se um corte virtual de cor cinza para a lateral do tanque, com a finalidade verificar o nível do líquido quando inserir animação à tela.

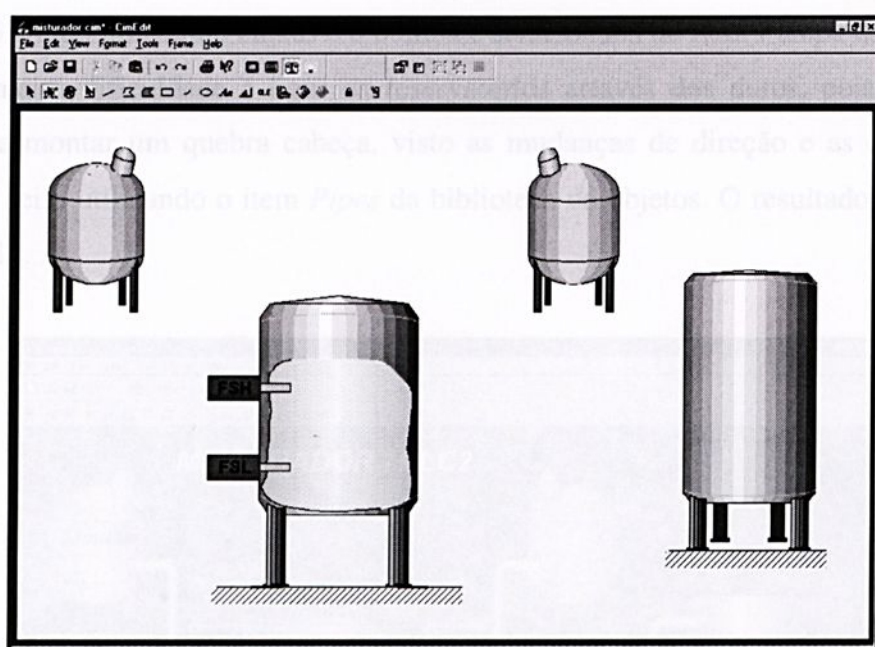


Figura C.16 - inserindo os sensores de nível baixo e alto.

Usando o *Text String* escreve-se a palavra “Depósito” e a insere sobre um dos depósitos. Para configurar o fundo da apresentação, basta pressionar, sem selecionar nenhum objeto, as teclas *Alt+Enter*, selecionando então a cor do fundo da tela, conforme a figura C.17.

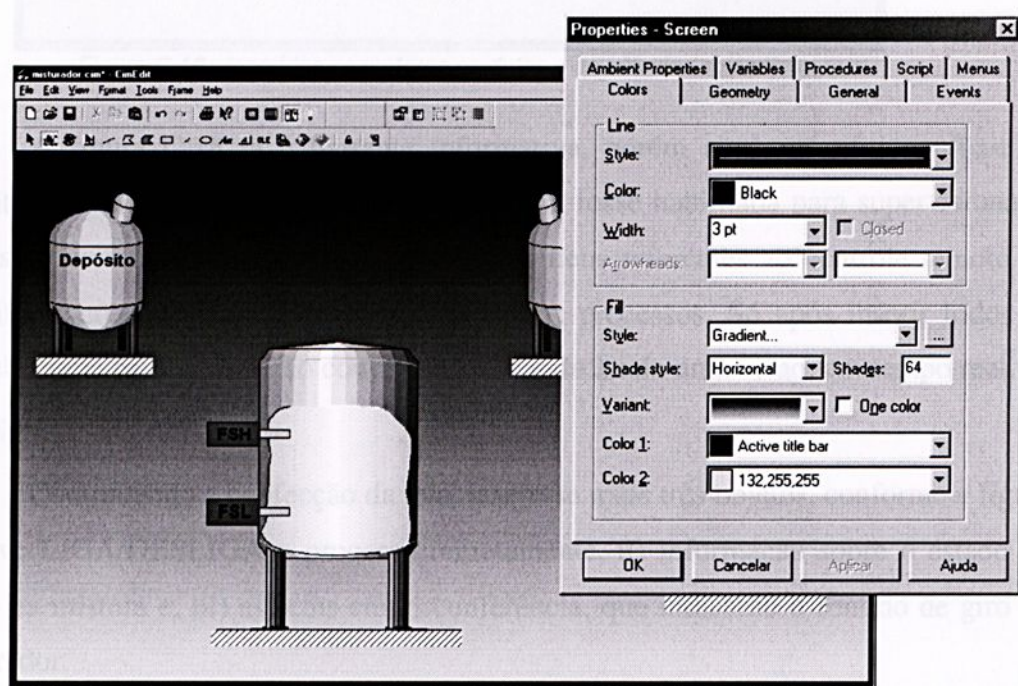


Figura C.17 - configurando o plano de fundo da tela.

O próximo passo é desenhar os dutos por onde passarão os líquidos, escrever o título da tela e inserir as válvulas solenóides e os motores. O título da tela é criado utilizando o *Text String*, as válvulas solenóides são representadas por quadrados vermelhos, o motor do

misturador é criado utilizando caixas e a bomba é selecionada no item *Pumps* da biblioteca de objetos. A maior dificuldade é ligar os reservatórios através dos dutos, pois essa tarefa é semelhante a montar um quebra cabeça, visto as mudanças de direção e as curvaturas dos dutos; isto é feito utilizando o item *Pipes* da biblioteca de objetos. O resultado é apresentado na figura C.18.

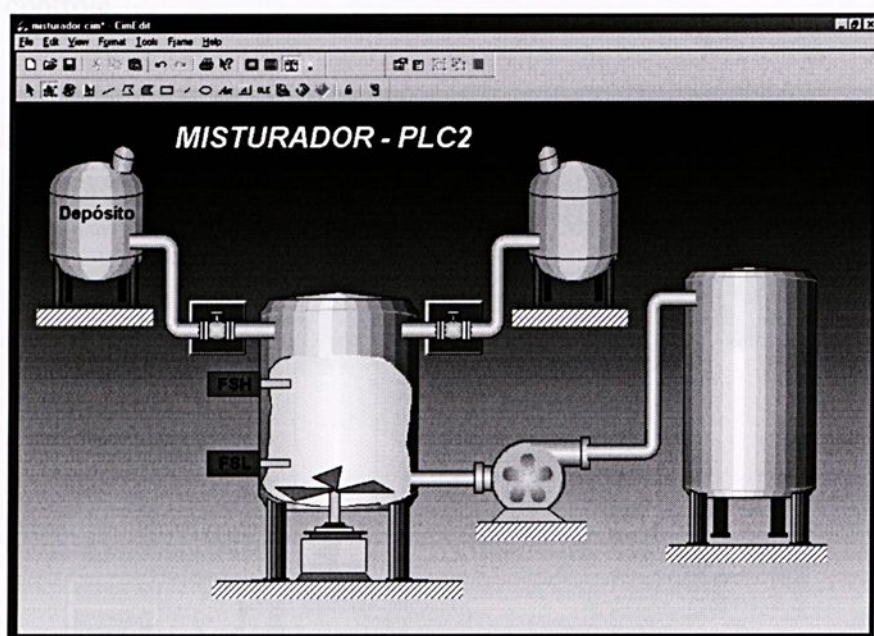


Figura C.18 - inseridos os motores os dutos e as válvulas solenóides.

A figura C.18 já é bastante informativa, porém ainda não foi configurado nenhum ponto na tela, ou seja, nada ocorreria se a tela fosse habilitada para supervisionar o sistema. Mesmo assim, ainda serão inseridos alguns itens referentes ao controle remoto do sistema e informações de intertravamento com outros processos. Só após inserir todos os objetos na tela é que os pontos serão configurados, dando à tela uma visão, em tempo real, de toda a operação.

Continuando a confecção da tela, insere-se mais três objetos, conforme a figura C.19: i) chave LIGA/DESLIGA o processo remotamente, ii) informação sobre o estado do motor usado na mistura e, iii) as setas em circunferência, que indicarão o sentido de giro do motor misturador.

A chave LIGA/DESLIGA é constituída de dois *Text Button*, um com fundo verde e outro com fundo vermelho, e mais um retângulo de fundo branco inserido atrás dos dois *Text Button*; todos objetos que compõe a chave foram agrupados. O estado do motor misturador é indicado por um *Text Button* vermelho e um *Text String* escrito “Desligado”, a

função deste é de informar quando o motor estiver ligado, mudando sua cor de vermelho para verde e informar também qual a configuração utilizada das bobinas do motor: estrela ou triângulo.

As setas ao lado do motor informarão o sentido de giro do mesmo. Todos esses objetos ainda deverão ser configurados para mostrarem os resultados desejados na tela de supervisão e controle.

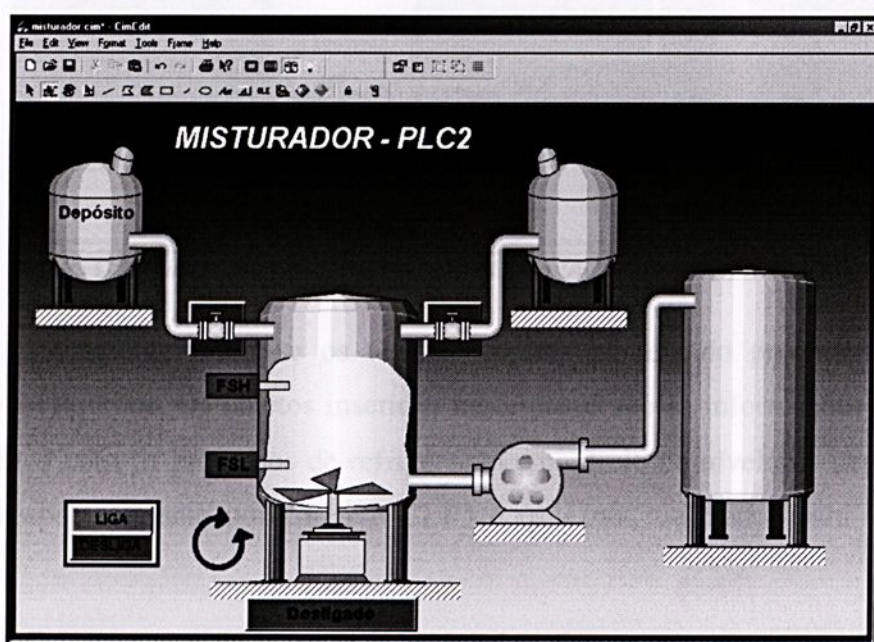


Figura C.19 - inseridos a chave LIGA/DESLIGA, e os indicadores de funcionamento do motor misturador.

Insere-se agora alguns arranjos na figura, preocupando-se sempre se a tela não ficará sobrecarregada de informações. Dos novos objetos inseridos (ver figura C.20), somente a vazão dos líquidos nos finais dos dutos dos depósitos e a frase “AQUECENDO A MISTURA” farão parte de um evento dinâmico, ficando os demais estáticos na tela. A razão de animar esses pontos é de mostrar que, estando as válvulas solenóides abertas, ocorrerá o enchimento do reservatório de mistura, e de mostrar o exato instante em que a etapa de aquecimento da mistura for iniciada.

A frase “AQUECENDO A MISTURA” é um *Text String*, que será configurado para aparecer somente durante a etapa de aquecimento e o a vazão é o agrupamento de várias gotas, selecionadas no item *Nature* da biblioteca de objetos.

O resultado da inserção desses novos objetos pode ser visto na figura C.20.

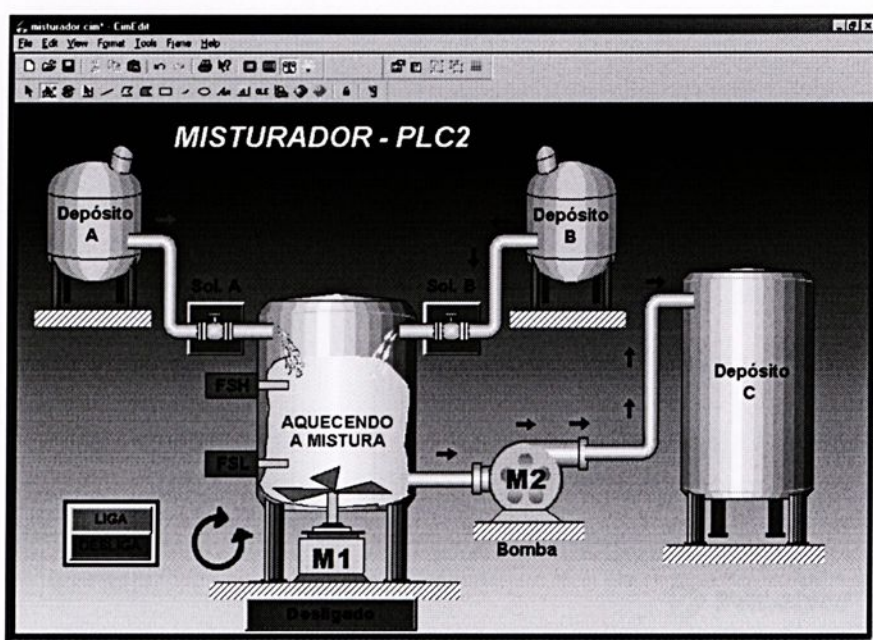


Figura C.20 - inseridos os arranjos e o Text String "AQUECENDO A MISTURA".

Finalmente insere-se os últimos objetos na tela do processo de mistura e aquecimento de líquidos. Os objetos inseridos neste passo são as informações referentes aos intertravamentos com os processos de refrigeração e controle de nível dos depósitos A e B, controlados respectivamente por CLP1 e CLP3 e os *Links*, utilizados para abrirem novas janelas.

O intertravamento com o processo de refrigeração mostra na tela o tempo em que a temperatura da água, que refrigera o motor M1, permanece acima do valor considerado limite. O intertravamento com o controlador de nível informa se o nível do líquido baixou demasiadamente, impedindo que a válvula solenóide seja aberta; este objeto será configurado para aparecer somente se ocorrer problema no nível do líquido.

Os *Links* são usados para abrir novas janelas, neste caso para ir à tela principal, contendo a visão geral de todo o processo e para abrir a tela contendo o gráfico de tendência da temperatura durante a etapa de aquecimento. Também foi inserida a informação do tempo decorrido da etapa de mistura, que foi programada para ocorrer durante 40 segundos.

Os resultados, de inserção de todos os objetos, são vistos na figura C.21. Deve-se agora configurar todos objetos para realizarem as tarefas propostas a cada um deles.

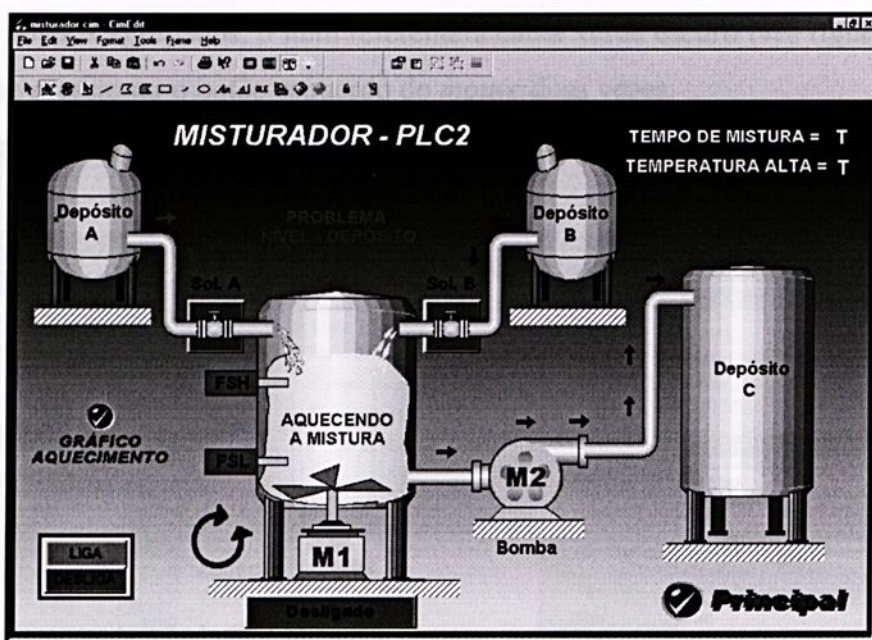


Figura C.21 - tela final de supervisão e controle do processo de mistura e aquecimento de líquidos.

C.1.2 Configurando os Objetos

Primeiramente será mostrado como configurar os pontos que mudarão de cor quando um evento acontecer. Os primeiros objetos que serão configurados são os sensores FSL e FSH, que deverão mudar de cor quando o nível do líquido passar por eles. Mas antes de começar a configuração dos objetos deve-se iniciar o projeto *Industria.gef*, através do *Workbench*, indo até o menu *Project* e pressionando em *Run*, ou simplesmente pressionando *Ctrl+R*.

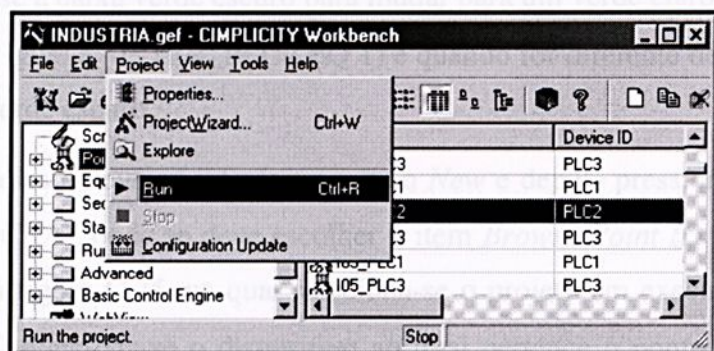


Figura C.22 - modo de iniciar um projeto.

Seleciona-se o objeto sensor FSL, que nada mais é senão o agrupamento entre a caixa verde escuro, a caixa cinza e o texto. Pressionando duas vezes o botão esquerdo do *mouse* sobre este objeto, abrirá a janela de propriedade dos objetos agrupados. Entre as várias guias da janela propriedades dos objetos agrupados, escolhe-se a guia *Group* e seleciona-se,

entre os três objetos agrupados, o item referente à caixa verde escuro (ver figura C.23), sobre o qual deve-se pressionar o botão esquerdo do *mouse* duas vezes.



Figura C.23 - abrindo a janela de propriedades do objeto sensor de nível baixo

Abrirá a nova janela de propriedades do objeto, na qual escolhe-se o item *Edit* da guia *Color Animation*. E é neste item que será configurada a mudança de cor quando algum ponto mudar seu valor. Neste caso, o sensor de nível baixo FSL está fisicamente ligado à entrada 4 do CLP2, logo o *bit* da memória %I04 é responsável por informar a situação deste sensor. Configura-se a caixa verde escuro para mudar para um verde claro sempre que o ponto %I04 do CLP2 for igual a “1” (I04_PLC2 EQ 1) e quando for diferente de “1” (I04_PLC2 NE 1) ele volta a cor verde escuro.

Para isso, deve-se selecionar o item *New* e depois pressionar sobre o botão de navegação (figura C.24), onde se deve escolher o item *Browse Point ID*. Abrirá outra janela, conforme mostra a figura C.25, na qual seleciona-se o projeto em execução (INDUSTRIA); no item *Device ID* seleciona-se o dispositivo ao qual pertence o ponto I04_PLC2, que é o dispositivo PLC2, e então pressiona-se o botão *Browse*, que mostrará todos os pontos configurados do CLP2. Utilizando a barra de rolagem, encontra-se o ponto I04_PLC2 e sobre ele deve-se pressionar o botão esquerdo do *mouse* duas vezes, configurando-o para atuar da forma desejada, como mostra a figura C.25.

Lembrando que o ponto I04_PLC2 deve ser o *Point ID*, configurado no *Workbench*, conforme mostrado nas figuras C.9 e C.10, e que represente a memória %I04 do CLP2 (ver figura C.26).

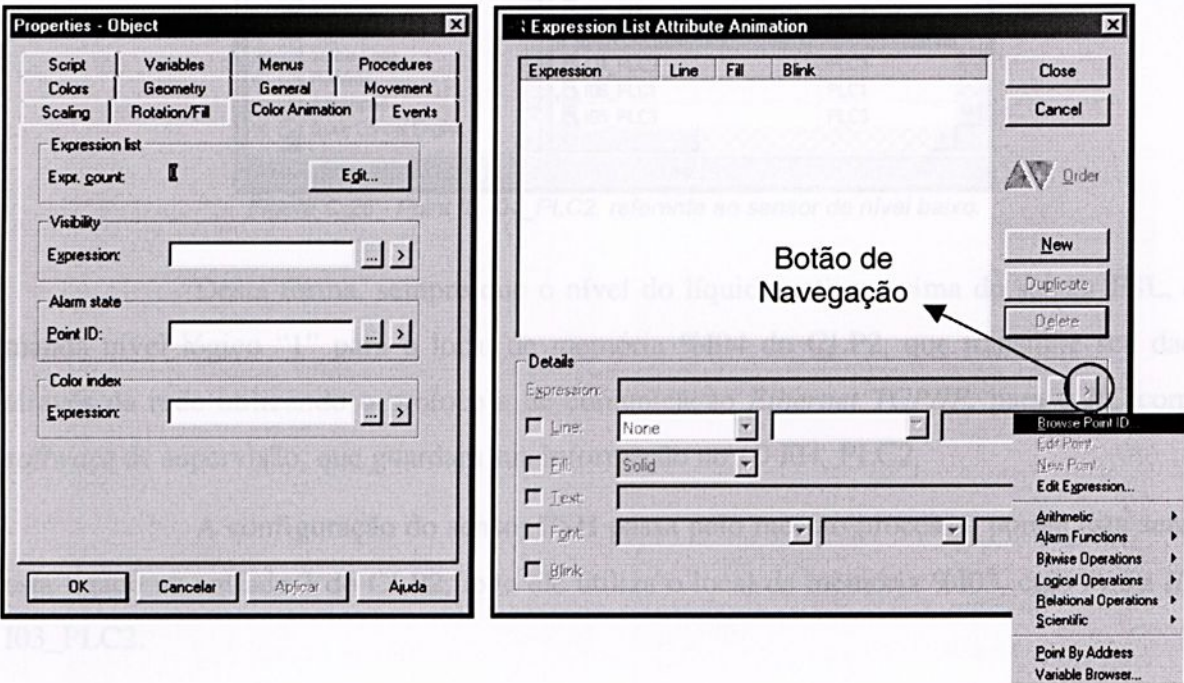


Figura C.24 - janelas para configuração da mudança de cor da caixa verde escura do sensor de nível baixo.

Nota-se na figura C.25, a configuração da caixa verde do sensor FSL, que muda de cor de acordo com o valor lógico do ponto I04_PLC2.

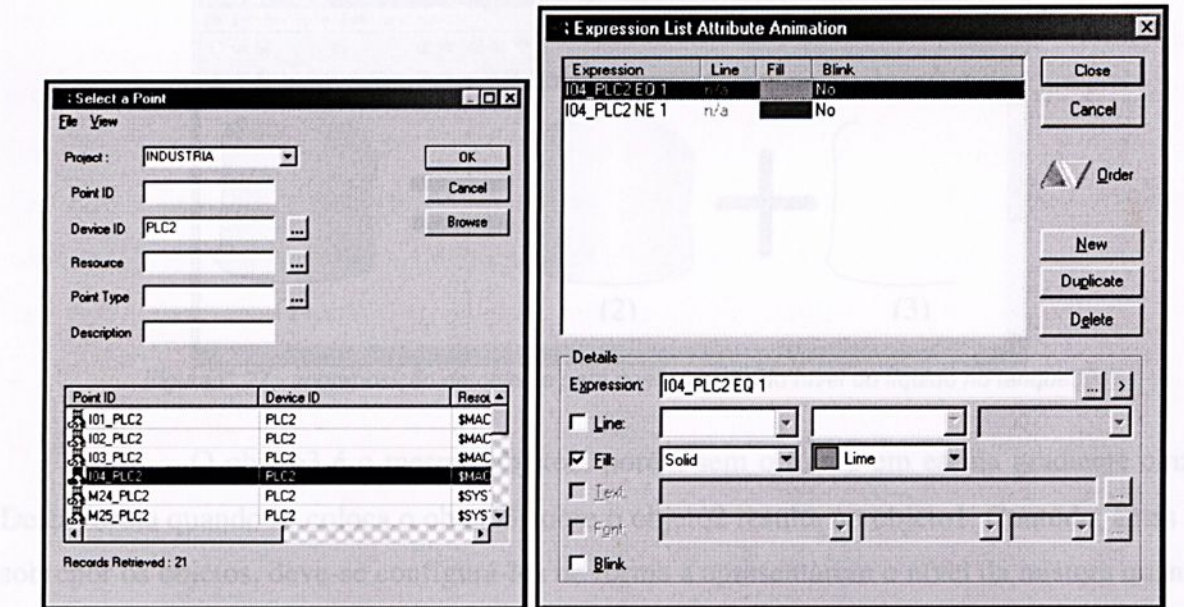


Figura C.25 - configurando a mudança de cor da caixa verde escura para verde claro.

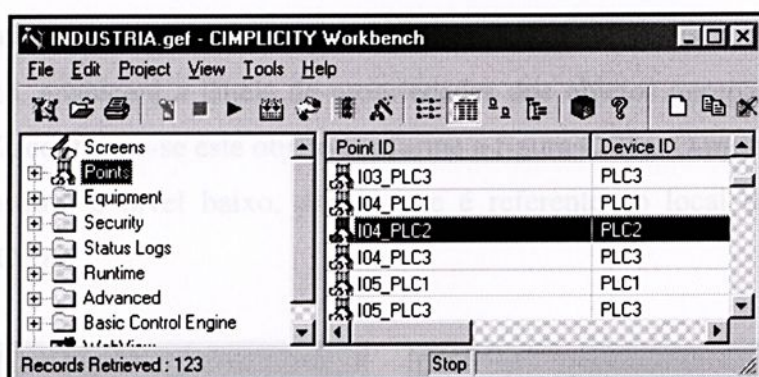


Figura C.26 - Point ID I04_PLC2, referente ao sensor de nível baixo.

Desta forma, sempre que o nível do líquido estiver acima do sensor FSL, ele manda nível lógico “1” para o local de memória %I04 do CLP2, que transmite seu dado, através da rede utilizando o protocolo de comunicação *Ethernet TCP/IP*, para o PC com o *software* de supervisão, que guardará sua informação no ID I04_PLC2.

A configuração do sensor FSH passa pelo mesmo processo, porém este sensor está ligado na entrada 3 do CLP2, logo ele utiliza o local de memória %I03, cujo *Point ID* é I03_PLC2.

Para configurar a animação do nível do líquido dentro do tanque, onde será realizada a mistura, utiliza-se sobreposição de objetos. A figura C.27 exemplifica como é feita a sobreposição de objetos.

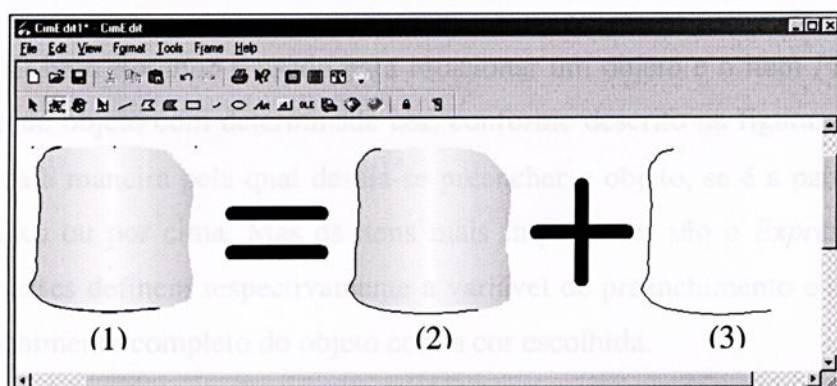


Figura C.27 - sobreposição de objetos para configuração do nível do líquido no tanque.

O objeto3 é o mesmo objeto2, porém sem o fundo em escala gradiente cinza. Desta forma quando se coloca o objeto3 sobre o objeto2 resulta no objeto1. Contudo, antes de sobrepor os objetos, deve-se configurá-los de forma a apresentarem o nível da mistura quando os sensores indicarem a presença do líquido. O objeto3 é configurado para indicar o nível alto (sensor FSH) e o objeto2 indica o nível baixo (sensor FSL). O nível do líquido se apresentará na cor verde claro e será utilizado outra ferramenta do *CimEdit* para criar esta animação.

Pressionando o botão esquerdo do *mouse* duas vezes sobre o objeto3, que indicará nível alto, aparecerá a janela de propriedades dos objetos agrupados. Escolhendo a guia *Rotation/Fill*, configura-se este objeto conforme a figura C.28a. O mesmo foi feito com o objeto2, que mostrará o nível baixo, porém este é referente ao local de memória %I04, conforme figura C.28b.

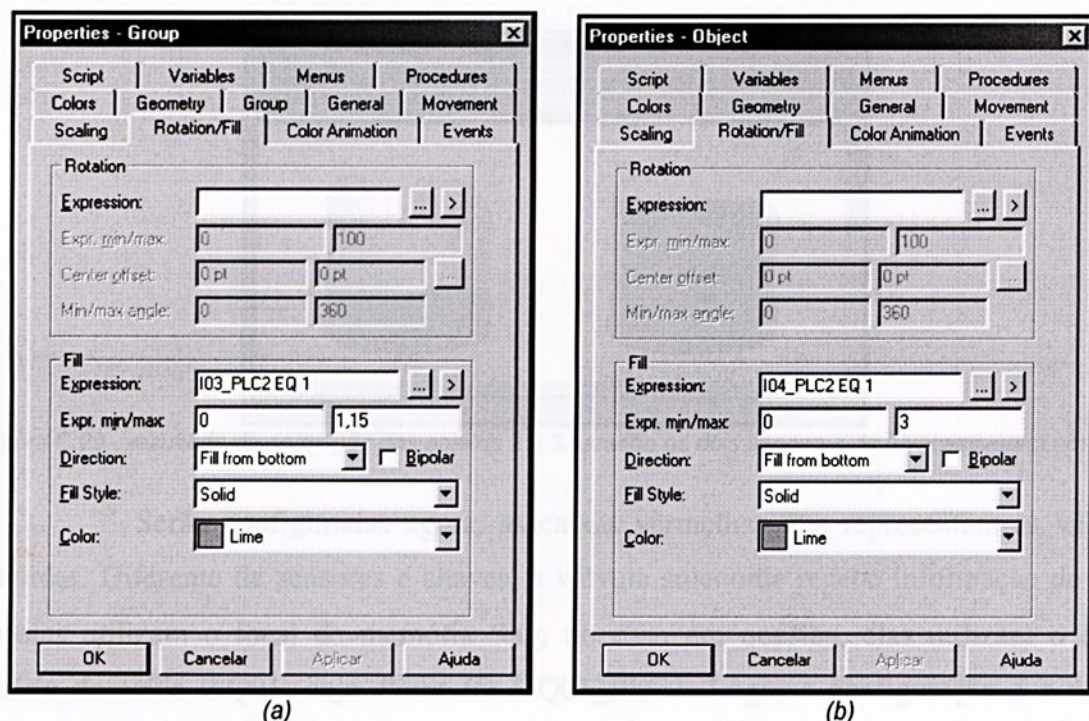


Figura C.28 - configuração da animação no reservatório de mistura - (a) objeto3, (b) objeto 2.

O item *Rotation* é usado para rotacionar um objeto e o item *Fill* é usado para preenchimento do objeto com determinada cor, conforme descrito na figura anterior. O item *Direction* indica a maneira pela qual deseja-se preencher o objeto, se é a partir da esquerda, direita, por baixo ou por cima. Mas os itens mais importantes são o *Expression* e o *Expr. min/max*, pois esses definem respectivamente a variável de preenchimento e o valor inicial e final para o enchimento completo do objeto com a cor escolhida.

Para indicar nível alto utilizou-se o mesmo sinal do sensor FSH (%I03), cujo *Point ID* é I03_PLC2 e pode receber o valor “0” ou “1”. Logo, deseja-se que o objeto3 da figura C.27 seja preenchido quase que totalmente, pela cor verde claro, sempre que o sensor de nível FSH fornecer nível lógico “1” para I03_PLC2. No item *Exp. min/max*, da figura C.28a, foi escolhido 0 (0%) para o mínimo e 1.15 (100%) para o máximo (enchimento completo do objeto3), assim quando I03_PLC2 receber o valor 1, o *software* preencherá o

objeto3 até 87% de todo o objeto, indicando a presença do líquido misturado na altura do sensor de nível alto.

O mesmo é feito para representar o nível baixo, mudando-se apenas o objeto, o *Point ID* e o valor a ser preenchido, que é 1/3 do objeto2 quando o *Point ID* I04PLC2 receber o valor 1. Assim sobrepondo os dois objetos resulta na animação desejada.

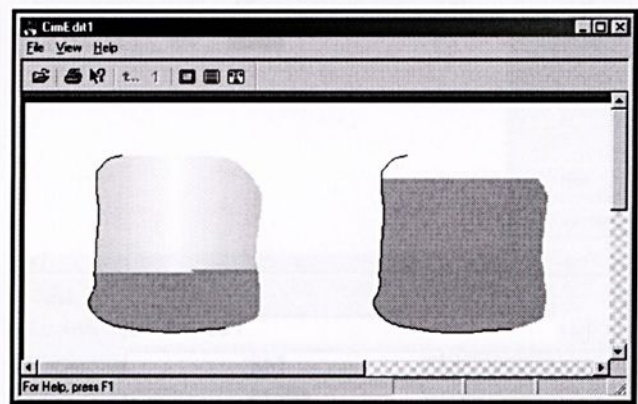


Figura C.29 - resultado da animação dos objetos 2 e 3, quando os dois sensores de nível estiverem em "1".

Serão configuradas agora, as caixas vermelhas que representam as válvulas solenóides. Diferente de sensores e chaves, a válvula solenóide recebe informação do CLP, logo elas utilizam o local de memória %Q; no caso em questão, elas utilizam o *bit* de memória de saída %Q01, cujo *Point ID* é Q01_PLC2. Logo, a configuração é a mesma utilizada para alteração de cor dos sensores (*Color Animation*), mudando apenas o *Point ID*, conforme descrito na figura C.30.

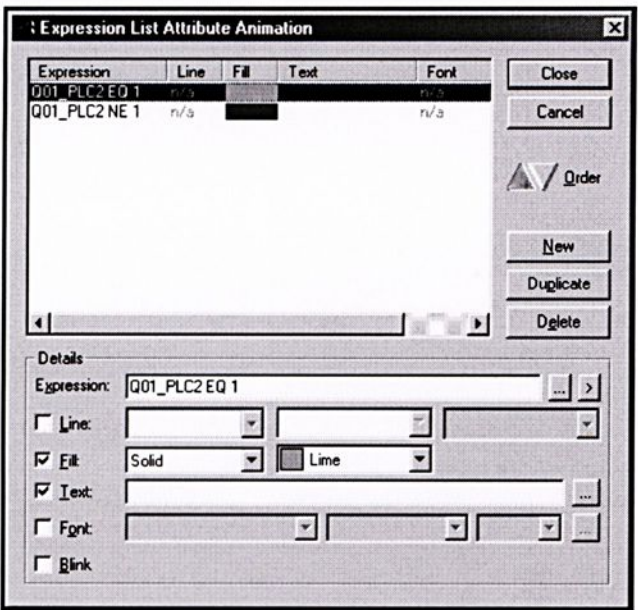


Figura C.30 - configurando a mudança de cor dos objetos referentes às válvulas solenóides.

A caixa vermelha abaixo do motor M1 também é configura utilizando o *Color Animation*. Ela ficará verde sempre que o motor M1, representado pelo *Point ID* Q07_PLC2, estiver em funcionamento. Dessa forma, a figura C.31 descreve a exata configuração desta caixa, para mudança de cor quando a saída %Q07 do CLP2 mudar seu estado lógico.

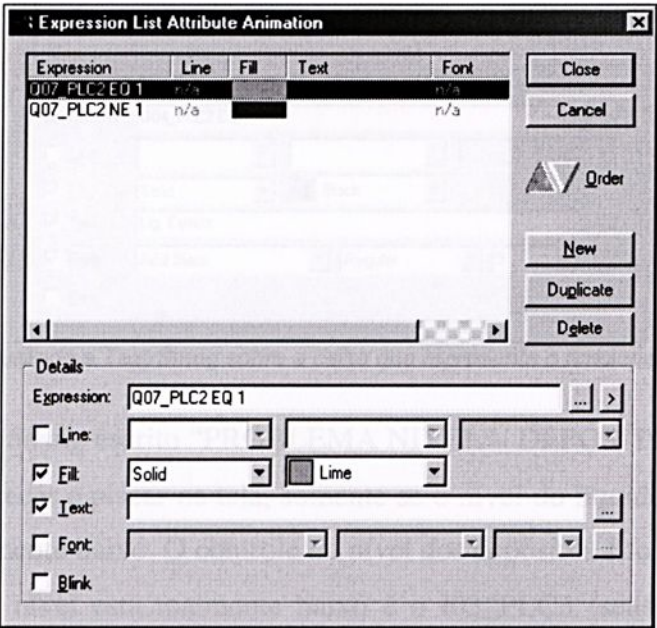


Figura C.31 - configurando a mudança de cor do objeto que representa o funcionamento do motor M1.

O *Text String*, sobre a caixa vermelha na figura C.21, escrito “Desligado” é configurado para mudar o texto, dependendo da configuração (estrela ou triângulo) em que o motor M1 estiver ligado. Também utiliza-se o *Color Animation* para criar esta animação. A figura C.32 mostra como configurar o *Text String* para mudar o texto quando os pontos Q04_PLC2 ou Q03_PLC2, referentes respectivamente a ligação estrela e triângulo, receberem o valor 1.

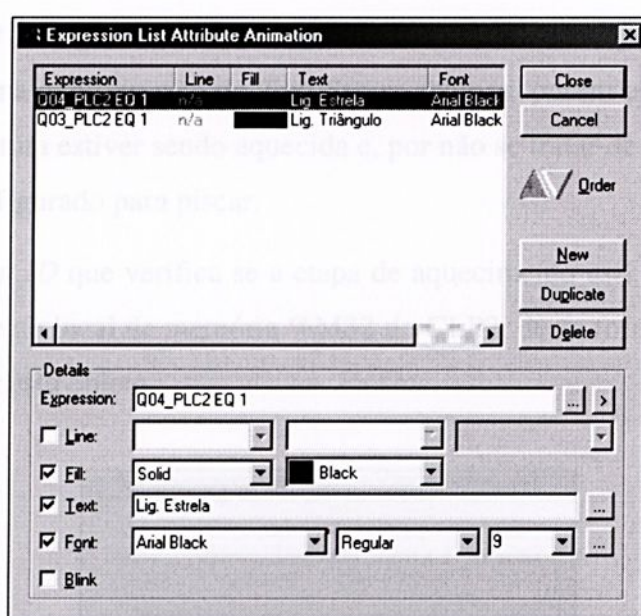
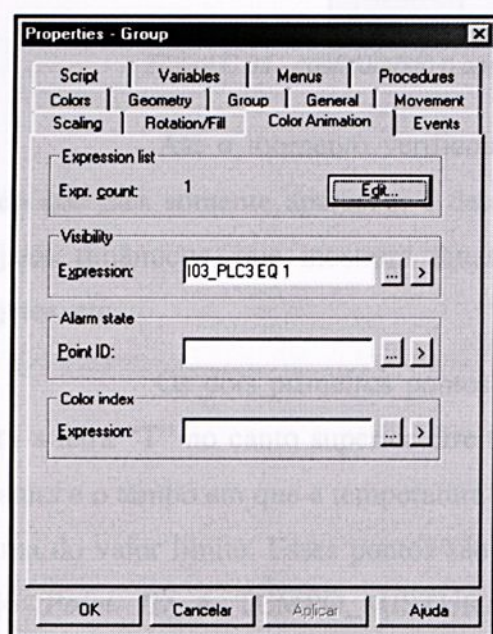
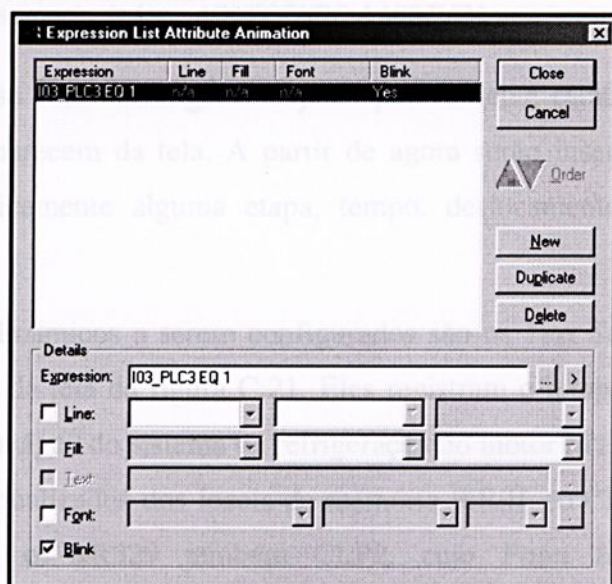


Figura C.32 - configurando a Text String sobre a caixa que representa o funcionamento do motor M1.

O Text String escrito “PROBLEMA NÍVEL – DEPÓSITO” (ver figura C.21) é configurado para aparecer e piscar na tela, somente se o nível do líquido dos depósitos A e B estiverem demasiadamente baixo. O controle do nível dos depósitos é feito no CLP3 e o ponto ID que informa se o nível está realmente baixo é o I03_PLC3 (sensor de nível baixo do processo controlado pelo CLP3). As figuras C.33a e C.33b mostram como configurar este objeto para visualizar o texto piscando na tela, somente em caso de nível baixo.



(a)



(b)

Figura C.33 - Alarmando problema no controlador de nível - (a) configurando o aparecimento da frase (b) janela referente ao botão Edit da guia Color Animation, que faz a frase piscar.

O *Text String* “AQUECENDO A MISTURA” (ver figura C.21) usa o mesmo artifício apresentado na configuração do *Text String* anterior, porém ele só aparece durante o período em que a mistura estiver sendo aquecida e, por não se tratar de um sinal de alerta, este *Text String* não é configurado para piscar.

O *Point ID* que verifica se a etapa de aquecimento está em funcionamento é o M32_PLC2, referente ao local de memória %M32 do CLP2, desta forma é este o ponto que é usado para configurar este objeto.

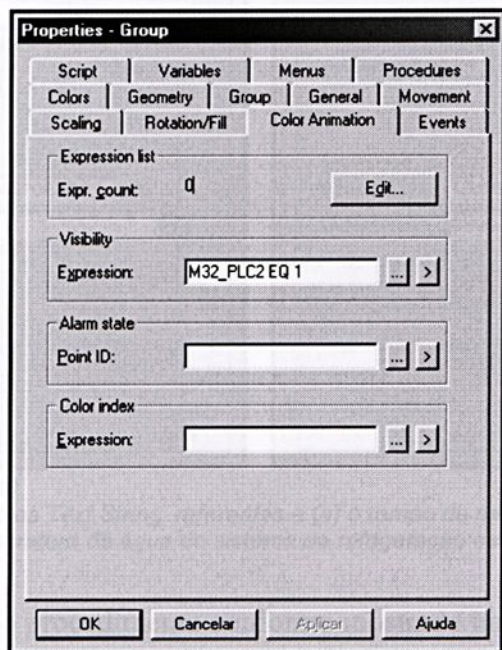


Figura C.34 - configurando o aparecimento da frase AQUECENDO A MISTURA.

Até o momento verificou-se como configurar objetos praticamente estáticos, dado que eles somente aparecem e desaparecem da tela. A partir de agora serão inseridos objetos dinâmicos, que mostram dinamicamente alguma etapa, tempo, deslocamento de objetos, etc.

Os dois primeiros pontos dinâmicos a serem configurados são os *Text String* com a letra “T” no canto superior direito da tela da figura C.21. Eles registram o tempo de mistura e o tempo em que a temperatura da água do sistema de refrigeração do motor M1 está acima do valor limite. Esses pontos são atualizados dos locais da memória %R01 do CLP2, cujo *Point ID* é TEMPO_MISTURA, e %R329 também CLP2, cujo *Point ID* é TEMPO_TEMP_ALTA.

Pressionando duas vezes o botão esquerdo do *mouse* sobre os objetos a serem configurados, abre-se a janela de propriedade, na qual deve-se escolher a guia *Text*, que será

configurada para mostrar, no lugar do texto, os tempos desejados, conforme inseridos no campo *Expression* do item *Display value* (figura C.35). Mostra-se nas figuras C.35a e C.35b que os *Point ID* estão sendo divididos por 10, pois na programação *Ladder* foi utilizado o bloco de função de tempo TMR na escala de 100 milissegundos.

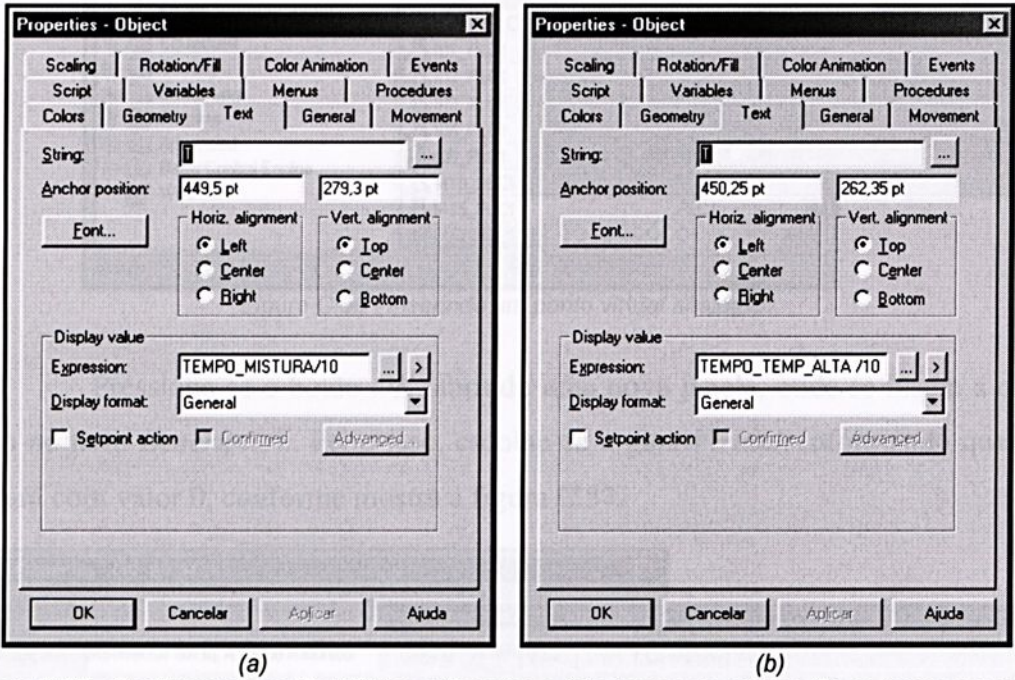


Figura C.35 - Configurando os *Text String*, referentes a (a) o tempo de mistura (b) ao tempo em que a temperatura da água do sistema de refrigeração está alta.

Executando o procedimento conforme mostrado na figura C.35, os *Text Strings* com a letra “T”, irão apresentar os tempos conforme desejado.

Por enquanto todos os objetos configurados na tela estão direto e unicamente relacionados com os dispositivos CLPs, utilizando locais de sua memória para efetuar alguma animação. Já em tempo, será apresentada a configuração de objetos dinâmicos, utilizando pontos virtuais, que são programados para funcionarem quando um evento do CLP acontecer. Esses objetos, após configurados, darão idéia, por exemplo, de movimento de líquido, sentido de giro, motor em funcionamento, etc.

Para isso, deve-se adicionar no *Workbench* os pontos virtuais que serão usados no *CimEdit*. O primeiro ponto apresentado é referente ao fluxo dos líquidos dos depósitos A e B para o reservatório onde será feita a mistura. Dá-se o nome para esse *Point ID* de LIQ_AB_MOVE, que será configurado no *Workbench* da mesma forma que pontos do dispositivo CLP, porém sendo agora, pontos virtuais. Para configurar o ponto virtual, primeiro pressiona-se o botão esquerdo do *mouse* duas vezes sobre o item *Points* do *Workbench*,



adicionando um novo ponto; digita-se o nome do ponto, por exemplo, LIQ_AB_MOVE, escolhendo ponto virtual de classe analógica, conforme apresentado na figura C.36.

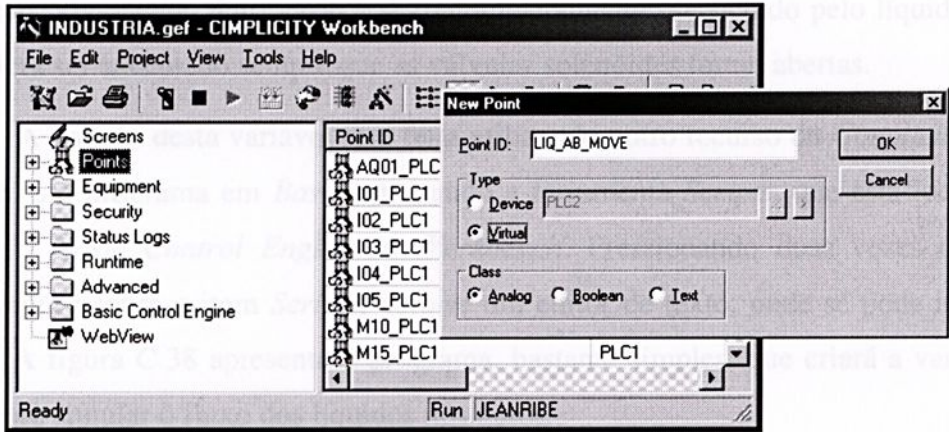


Figura C.36 - inserindo um ponto virtual analógico.

Pressiona-se o botão *OK*, abrindo uma nova janela, onde se insere a descrição do ponto no item *Description*. Feito isso, escolhe-se a guia *Virtual*, informando que o ponto inicializará com valor 0, conforme mostra a figura C.37.

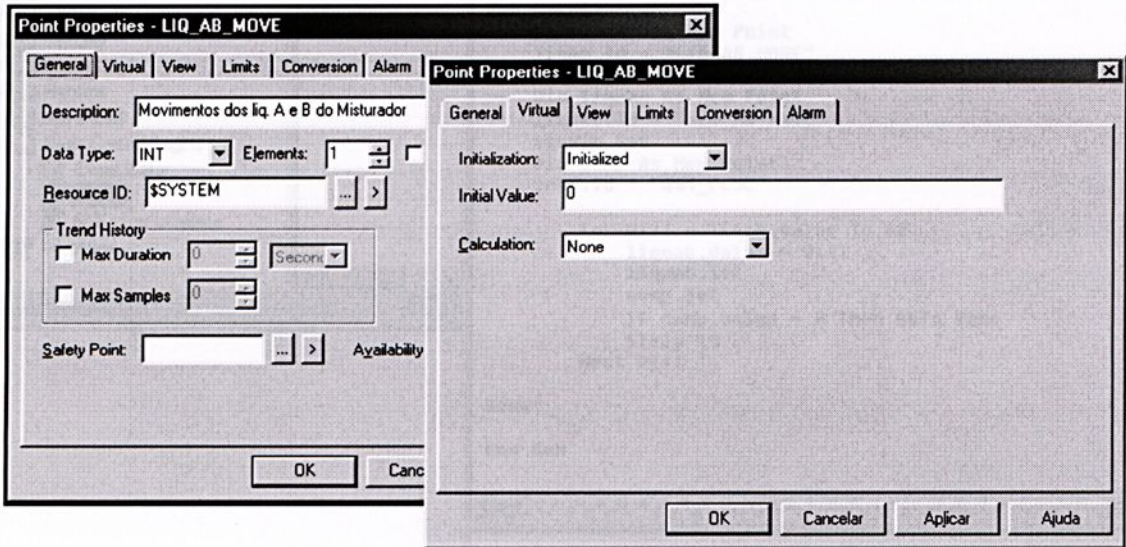


Figura C.37 - configurando as propriedades do ponto virtual analógico.

Após isto, este ponto virtual estará adicionado no projeto, logo pode-se utilizá-lo no *CimEdit*. Para criar o movimento dos líquidos A e B, dos depósitos até o tanque, onde será realizada a mistura, utiliza-se a ferramenta *Rotation/Fill*, mesma utilizada para mostrar o nível do tanque. Desta vez, os objetos que estarão sendo preenchidos (*Fill*) serão os dutos por onde passam os líquidos. O *Point ID* LIQ_AB_MOVE terá a tarefa de informar qual a posição virtual do líquido nos dutos.

Deve-se então escrever um programa que transfira para o *Point ID* LIQ_AB_MOVE uma variável, com pequeno passo, de zero até um valor máximo, simulando, respectivamente, duto vazio e duto completamente preenchido pelo líquido. Esse programa deverá ser executado sempre que as válvulas solenóides forem abertas.

A criação desta variável será feita utilizando outro recurso do *Cimplicity HMI*, que é escrever um programa em *Basic*, utilizando a ferramenta *Scripts*, que está localizada dentro da pasta *Basic Control Engine* do *Workbench*. Pressionando duas vezes o botão esquerdo do *mouse* sobre o item *Scripts*, abre-se um editor de texto, onde se pode iniciar a programação. A figura C.38 apresenta o programa, bastante simples, que criará a variável a ser utilizada para simular o fluxo dos líquidos nos dutos.

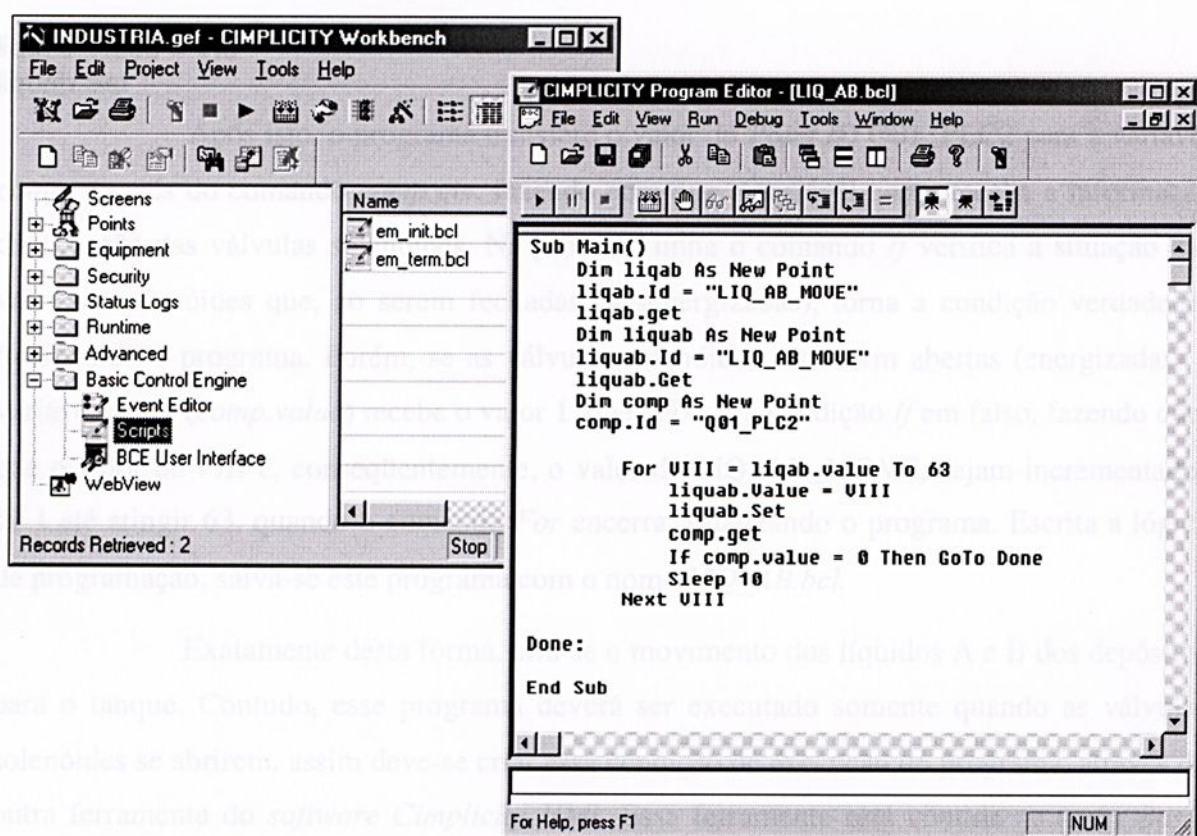


Figura C.38 - Script de programação para mudar o valor do ponto virtual.

As funções utilizadas, para essa programação, são muitas conhecidas (*For*, *While*, *If*, etc). Na figura C.38, *Sub Main()* e *End Sub*, informam o início e o final do programa. As funções *Dim* são usadas para declarar variáveis, neste caso declara-se que os pontos *liqab* e *liquab* representam o *Point ID* LIQ_AB_MOVE (ponto virtual analógico) e *comp* representa Q01_PLC2 (válvula solenóide).

A função *.get*, depois da variável, faz com que a variável receba o valor atual do seu *Point ID* e a função *.set*, transfere o valor da variável para o *Point ID* correspondente. Assim, logo após as declarações das variáveis, *liqab* e *liquab* recebem o valor atual de *LIQ_AB_MOVE*, enquanto que a variável *comp* não recebe valor algum logo após sua declaração.

A função *For* incrementa em um a variável *VIII* do valor inicial de *liqab* (*liqab.value*) até 63, que será o máximo valor de preenchimento dos dutos, que levam os líquidos A e B dos depósitos para o tanque de mistura.

O valor da variável *VIII* é movido para a variável *liquab* e logo após é transferida para o *Point ID* *LIQ_AB_MOVE*, através dos comandos,

```
liquab.Value = VIII  
liquab.Set
```

Após isto, o programa transfere o valor do *Point ID* *Q01_PLC2* para a variável *comp*, através do comando *comp.get*, fazendo com que a variável *comp* receba a informação da situação das válvulas solenóides. Na próxima linha o comando *If* verifica a situação das válvulas solenóides que, ao serem fechadas (desenergizadas), torna a condição verdadeira, finalizando o programa. Porém, se as válvulas solenóides estiverem abertas (energizadas) a variável *comp* (*comp.value*) recebe o valor 1, que deixará a condição *If* em falso, fazendo com que o valor de *VIII* e, conseqüentemente, o valor de *LIQ_AB_MOVE*, sejam incrementados de 1 até atingir 63, quando o comando *For* encerra, finalizando o programa. Escrita a lógica de programação, salva-se este programa com o nome *LIQ_AB.bcl*.

Exatamente desta forma, cria-se o movimento dos líquidos A e B dos depósitos para o tanque. Contudo, esse programa deverá ser executado somente quando as válvulas solenóides se abrirem, assim deve-se criar essa condição de execução do programa, através de outra ferramenta do *software Cimplicity HMI*. Essa ferramenta está contida na pasta *Basic Control Engine* do *Workbench*, com o nome de *Event Editor* ou Editor de Eventos.

Para entender o funcionamento do *Event Editor*, será configurado a condição de execução do programa *LIQ_AB.bcl*. Este programa deverá ser executado somente se as válvulas solenóides estiverem abertas, ou seja, se o valor do *Point ID* *Q01_PLC2* for igual a 1.



Primeiro pressiona-se duas vezes o botão esquerdo do *mouse* sobre o item *Event Editor*, que abrirá a janela do editor de eventos. Feito isso, pressiona-se o botão *New* ou as teclas *Crtl+Enter* para criar um novo evento, ao qual dá-se o nome de *LIQ_AB_SOL*.

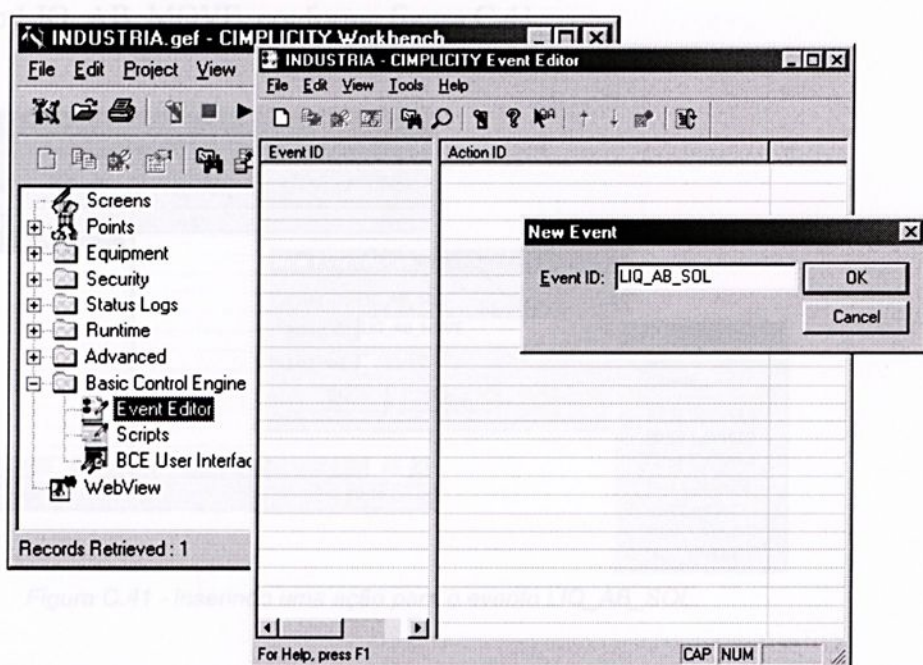


Figura C.39 - inserindo um novo evento ao projeto.

Pressionando o botão *OK*, será aberta a nova janela, onde será configurado o evento a ser verificado; neste caso, em particular, será verificada a situação da válvula solenóide, cujo *Point ID* é *Q01_PLC2*. Na guia *Advanced* escolhe-se o modo de execução paralela.

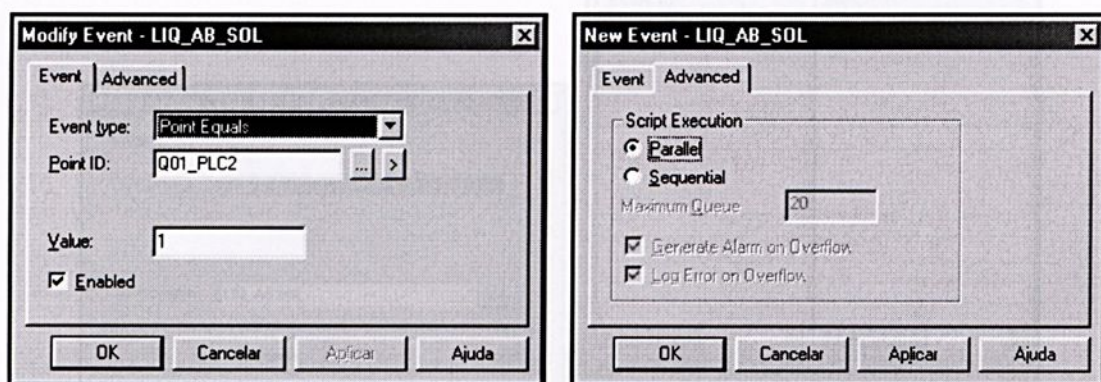


Figura C.40 - configurando o evento para acontecer sempre que o *Point ID* *Q01_PLC2* estiver com valor 1.

Todo evento configurado no editor de eventos tem alguma ação, ou seja, no caso apresentado, quando o editor de eventos verificar que o *Point ID* é igual a 1, executa-se o *Script LIQ_AB.bcl*. Portanto deve-se agora configurar a ação, ou ações, que o evento *LIQ_AB_SOL* irá executar quando o *Point ID* *Q01_PLC2* for 1.

Para isso, pressiona-se o botão esquerdo do *mouse* duas vezes sobre uma linha da coluna *Action ID* (ver figura C.41), abrindo a janela *New Event-Action*, onde se escolhe um nome para a ação. Pressionando o botão de navegação, escolhe-se a opção *New*, para criar o novo ponto de ação LIQ_AB_MOVE, conforme figura C.41.

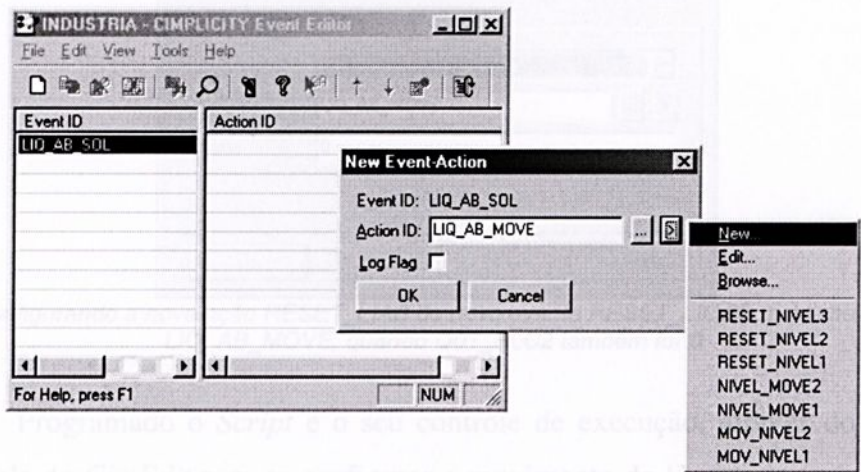


Figura C.41 - inserindo uma ação para o evento LIQ_AB_SOL.

Após configurar o novo ponto de ação, deve-se informar qual ação o editor de evento tomará. Neste caso, ele deverá executar um *Script* (figura C.42a); assim ao se pressionar o botão *OK*, configura-se a próxima janela como mostra a figura C.42.

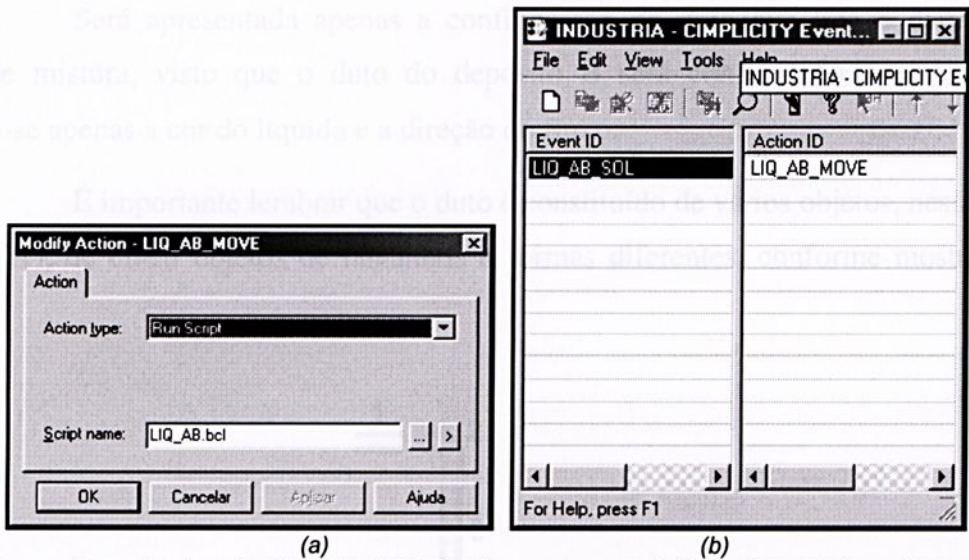


Figura C.42 - configuração da ação LIQ_AB_MOVE - (a) executar o Script LIQ_AB.bcl, (b) resultado da inserção do evento e da ação.

Para completar o procedimento de criação da variável que simulará o movimento dos líquidos, configura-se mais um evento chamado RESET_LIQAB, cuja ação será transferir para o *Point ID* LIQ_AB_MOVE o valor 0, quando as válvulas solenóides



fecharem, ou seja, quando o valor de Q01_PLC2 for 0. Assim não haverá simulação de fluxo de líquido dos depósitos para o tanque quando as válvulas estiverem fechadas (desenergizadas).

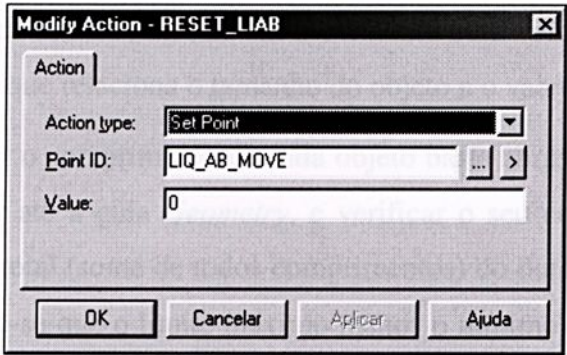


Figura C.43 - configurando a nova ação RESET_LIAB do novo evento RESET_LIQAB, para mover o valor 0 para LIQ_AB_MOVE, quando Q01_PLC2 também for 0.

Programado o Script e o seu controle de execução, através do Event Editor, retorna-se à tela do CimEdit para se configurar o movimento do líquido através dos dutos que ligam os depósitos ao tanque de mistura.

O movimento ocorrerá de acordo com a variável LIQ_AB_MOVE, que varia de 0 à 63. Desta forma, estes serão os limites mínimo e máximo do Expr. min/Max do item Fill. Será explicado em etapas como configurar os dutos.

Será apresentada apenas a configuração do duto que liga o depósito A ao tanque de mistura, visto que o duto do depósito B será configurado da mesma forma, mudando-se apenas a cor do líquido e a direção do fluxo.

É importante lembrar que o duto é constituído de vários objetos, neste caso (ver figura C.21), de cinco objetos de tamanhos e formas diferentes, conforme mostra a figura C.44.

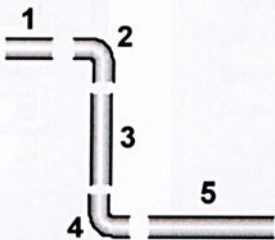


Figura C.44 – objetos desagrupados, que compõem o duto que liga o depósito A ao tanque de mistura.

O líquido deverá se mover no primeiro objeto da esquerda para direita, no segundo e terceiro objetos ele se move de cima para baixo, no quarto e quinto objetos o

líquido moverá da esquerda para direita novamente. Assim, deve-se agora distribuir o valor da variável de preenchimento em cada objeto, tendo-se certeza apenas, que o valor mínimo do primeiro objeto é 0, e o valor máximo do último objeto que é 63. Intuitivamente pode-se estimar o valor máximo e mínimo da variável LIQ_AB_MOVE em cada objeto, porém, desejando-se um resultado melhor, cria-se um movimento com velocidade constante, usando apenas uma regra de três que relaciona o tamanho do objeto e o valor que ele assume.

Para medir o comprimento de cada objeto basta abrir a janela de propriedades, do objeto em questão, ir até a guia *Geometry*, e verificar o seu comprimento. No caso em questão, o comprimento total (soma de todos comprimentos) do duto é 126pts¹, assim através de uma regra de três sabe-se que o limite máximo menos o mínimo, que se deve utilizar para configurar o preenchimento, de cada objeto é exatamente a metade de seu comprimento.

Desta maneira foram configurados os objetos que compõem o duto que liga o depósito A até o tanque de mistura, utilizando o *Rotation/Fill* da janela de propriedades do objeto do *CimEdit*. As figuras a seguir mostram a configuração de cada um dos cinco elementos, que compõe o duto.

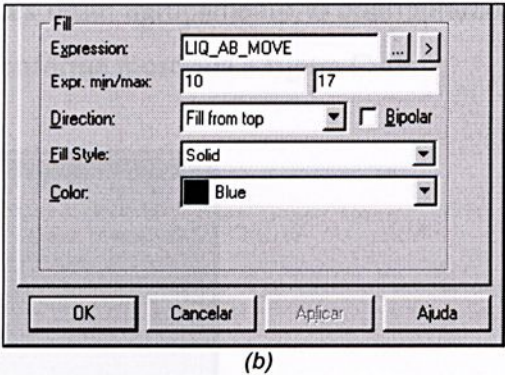
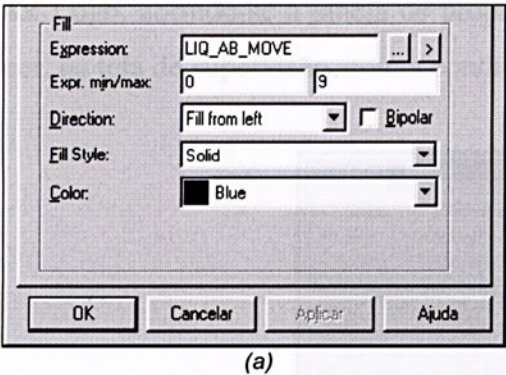


Figura C.45 - configuração do *Rotation/Fill* para preenchimento de objeto – (a) objeto1, (b) objeto2.

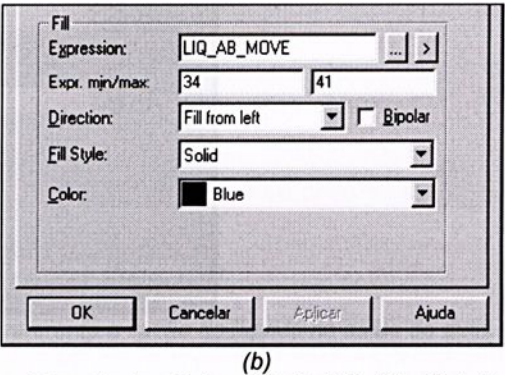
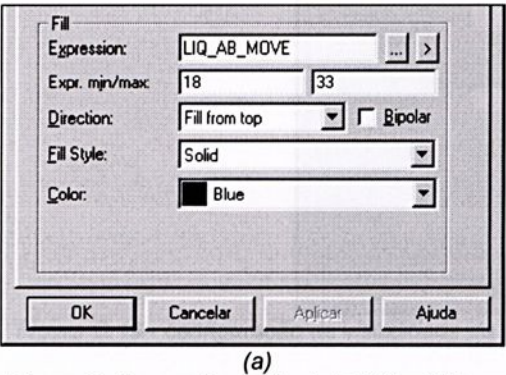


Figura C.46 - configuração do *Rotation/Fill* para preenchimento de objeto – (a) objeto3, (b) objeto4.

¹ Por essa razão o *Point ID* LIQ_AB_MOVE foi configurado para variar até 63.



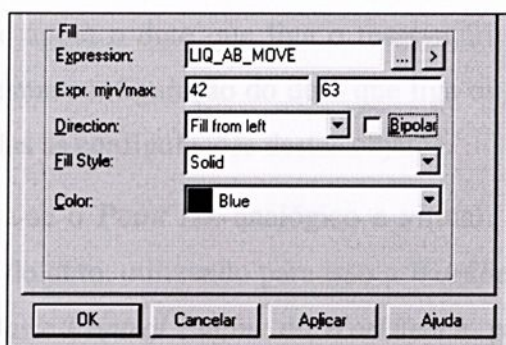


Figura C.47 - configuração do Rotation/Fill para preenchimento do objeto5.

Após configurado o fluxo dos líquidos nos dutos, será configurado outro objeto, que aparecerá somente se o fluxo dos líquidos chegar até a parte final do duto, ou seja, somente se LIQ_AB_MOVE for 63.

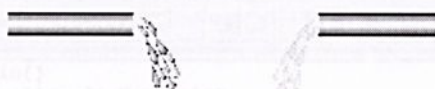


Figura C.48 - Vazão dos líquidos na saída dos dutos.

O objeto em questão é a vazão do líquido na saída do duto, representado pelo agrupamento de gotas azuis e amarelas (ver figura C.48), cada um referente ao líquido em questão. Logo, abrindo-se a janela de propriedades deste agrupamento, o configuramos para aparecer, na tela de supervisão, com a condição conforme mostrada a figura C.49.

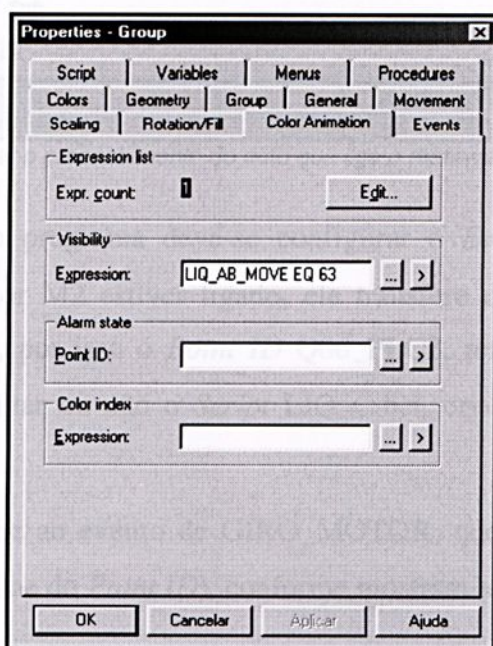


Figura C.49 - configuração de visibilidade do objeto referente a vazão do líquido na saída do duto.

Configura-se agora o duto que liga o reservatório de mistura ao depósito C. Como se trata quase da mesma configuração do duto que liga os depósitos ao reservatório de mistura, não serão detalhadas as configurações destes objetos.

Primeiro cria-se o *Point ID*, analógico e virtual, LIQ_C_MOVE, referente à variável de preenchimento do duto, utilizando para isso o *Workbench*. Utiliza-se a ferramenta *Script*, presente na pasta *Basic Control Engine* do *Workbench*, para programar uma variável que variará de 0 à 83, conforme mostra a figura C.50. Salva-se este programa com o nome LIQ_C.bcl.

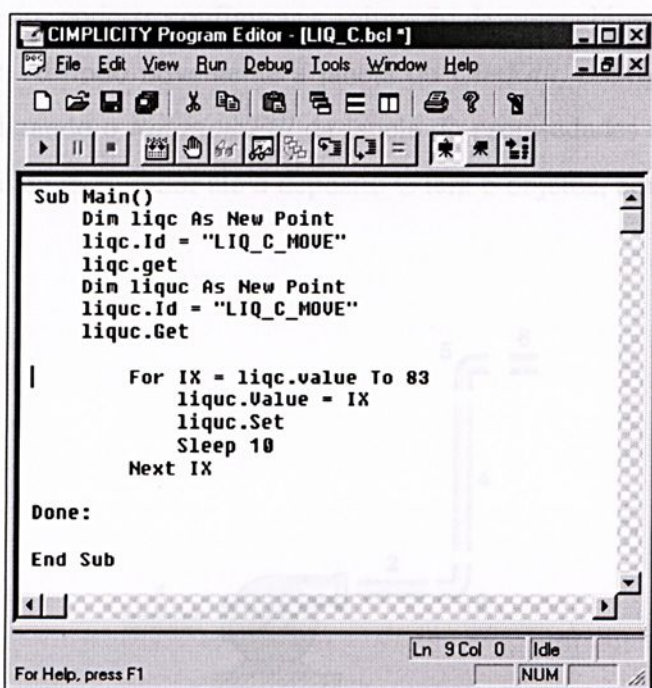
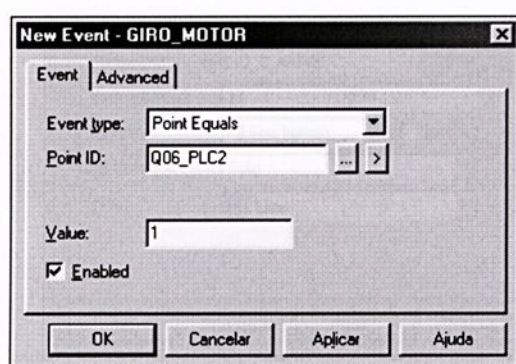


Figura C.50 - Script para criar o preenchimento do duto que liga o reservatório de mistura ao depósito C.

Concluído o programa deve-se configurar o *Event Editor* para executar o *Script*. Sempre que o motor M2 estiver ligado, ele transfere o líquido do reservatório de mistura para o depósito C, por isso o *Point ID* Q06_PL2, referente ao motor M2 será o ponto analisado para executar ou não o *Script* LIQ_C.bcl, criado para gerar a variável do ponto LIQ_C_MOVE.

Dá-se o nome ao evento de GIRO_MOTOR, que executará a ação de nome LIQ_C_MOVE (mesmo nome do *Point ID*), conforme mostram as figuras a seguir.



(a)



(b)

Figura C.51 - configurando um novo evento e uma nova ação - (a) evento GIRO_MOTOR, (b) ação LIQ_C_MOVE.

Após isto, pode-se configurar a animação de preenchimento de líquido no duto utilizando o *CimEdit*. Abrindo novamente a janela do *CimEdit*, configura-se os objetos que compõem o duto, utilizando o *Rotation/Fill* da janela de propriedades do objeto. Desta vez o duto que liga o tanque do misturador até o depósito C tem 6 objetos, conforme mostra a figura C.52.

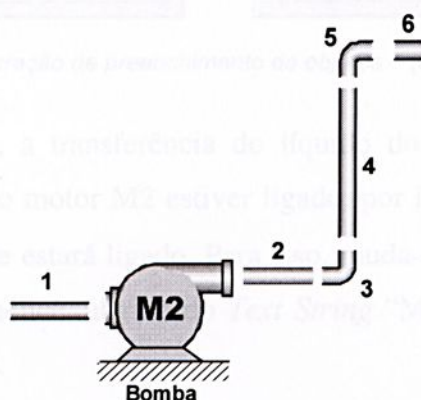
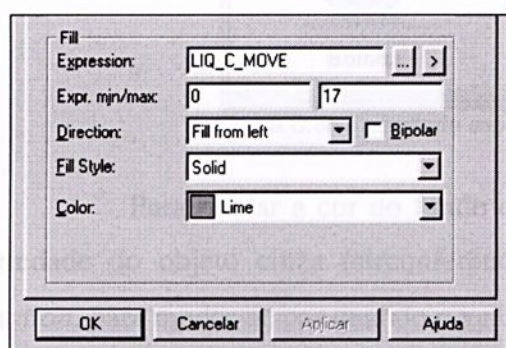
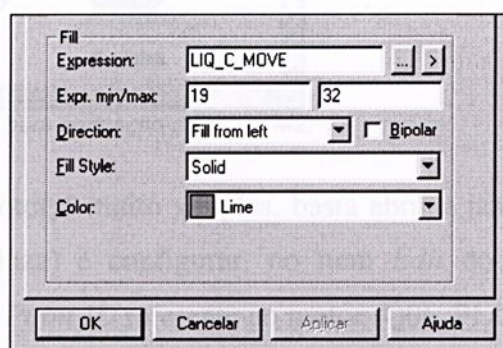


Figura C.52 – objetos desagrupados, referentes ao duto que liga o reservatório de mistura ao depósito C.

As configurações de cada objeto são apresentadas conforme mostram as figuras a seguir.

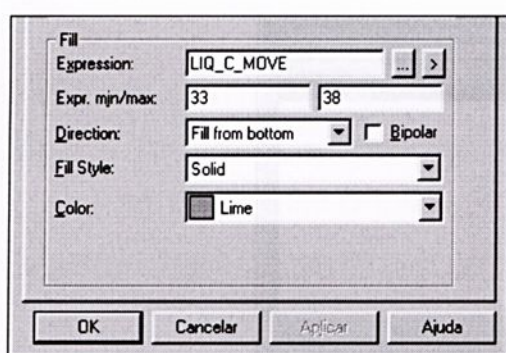


(a)

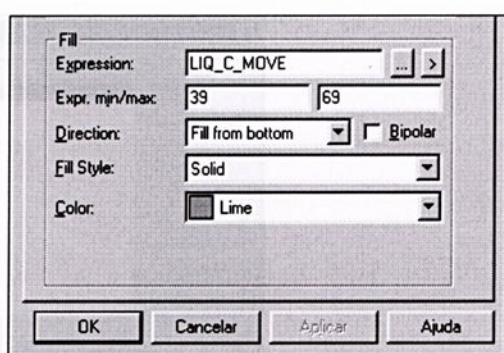


(b)

Figura C.53 - configuração de preenchimento de objetos - (a) objeto1, (b) objeto2.

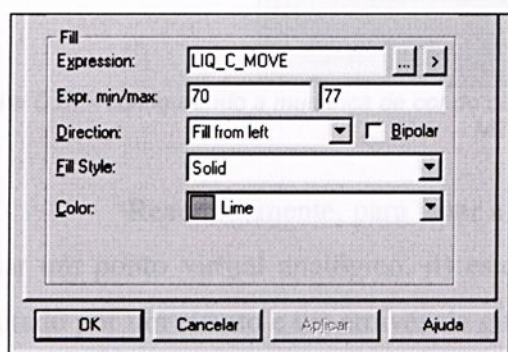


(a)

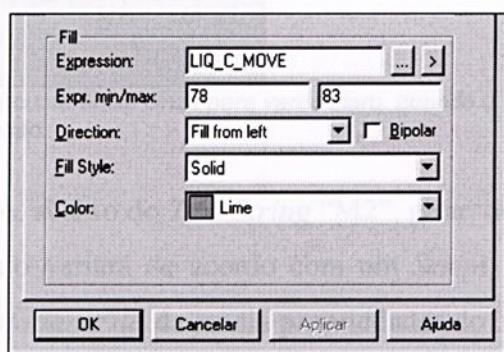


(b)

Figura C.54 - configuração de preenchimento de objetos – (a) objeto3, (b) objeto4.



(a)



(b)

Figura C.55 - configuração de preenchimento de objetos – (a) objeto5, (b) objeto6.

Como foi dito, a transferência do líquido do tanque do misturador para o depósito C só ocorre quando o motor M2 estiver ligado, por isso o motor será animado para indicar os instantes em que ele estará ligado. Para isso, muda-se a cor de fundo do motor, de cinza para verde e gira-se a hélice, abaixo do *Text String* “M2”, sempre que o *Point ID* do motor indicar nível lógico “1”.

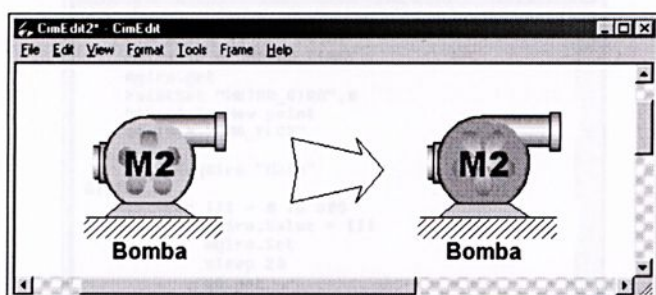


Figura C.56 - resultado esperado para animação do motor M2.

Para mudar a cor do fundo do motor é muito simples, basta abrir a janela de propriedade do objeto cinza (circunferência cinza) e configurar, no item *Edit* do *Color Animation*, para mudar de cor quando o valor do *Point ID* referente ao motor (Q06_PLC2) for igual a 1, conforme mostra a figura C.57.

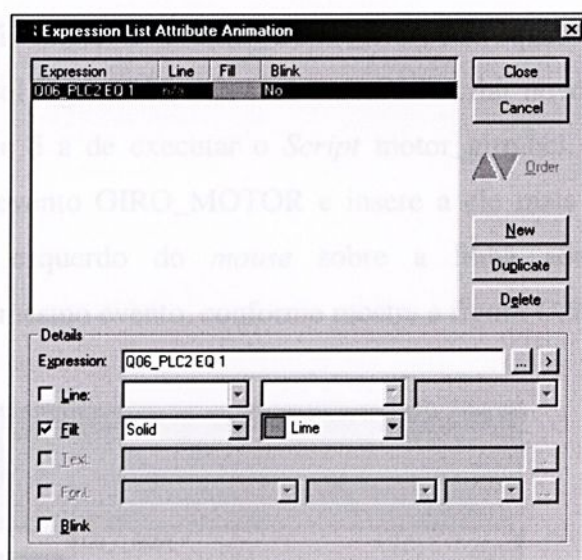


Figura C.57 - configurando a mudança de cor do objeto circunferência cinza para verde claro, quando o motor M2 for ligado.

Resumidamente, para fazer a hélice, abaixo do *Text String* “M2”, girar, deve-se i) criar um ponto virtual analógico, ii) este ponto variará de acordo com um *Script*, que é executado por um evento e iii) através da guia *Rotation/Fill*, da janela propriedades do objeto, dá-se ao objeto o movimento desejado.

Utilizando o item *Points* do *Workbench*, cria-se o *Point ID*, virtual e analógico, MOTOR_GIRO. O programa que faz este ponto variar é escrito na ferramenta de edição de programas *Script* e recebe o nome de giro_motor.bcl, conforme mostra a figura C.58.

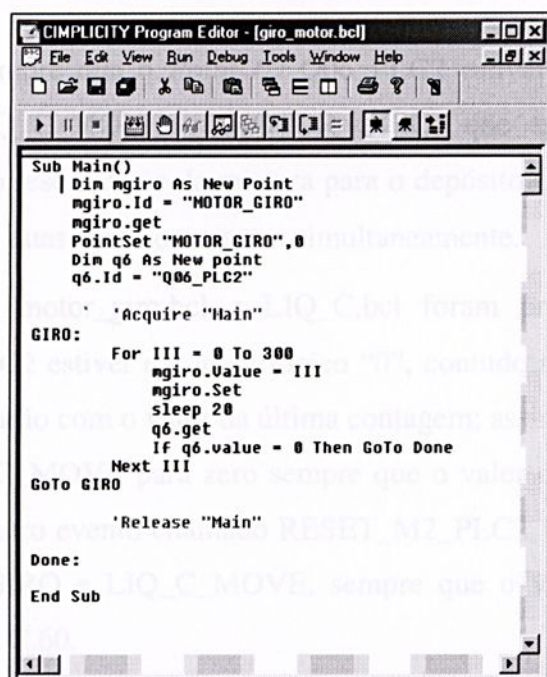


Figura C.58 - Script utilizado para variar o ponto que fará a hélice, do motor M2, girar.

Como já existe o evento GIRO_MOTOR, que verifica se o *Point ID* Q06_PLC2 está em nível lógico “1”, não será necessário criar novo evento, acrescentando a ele mais uma ação, que é a de executar o *Script* motor_giro.bcl. Assim, dentro do *Event Editor*, seleciona-se o evento GIRO_MOTOR e insere a ele mais uma ação, pressionando duas vezes o botão esquerdo do *mouse* sobre a linha abaixo da ação existente (LIQ_C_MOVE) deste mesmo evento, conforme mostra a figura C.59.

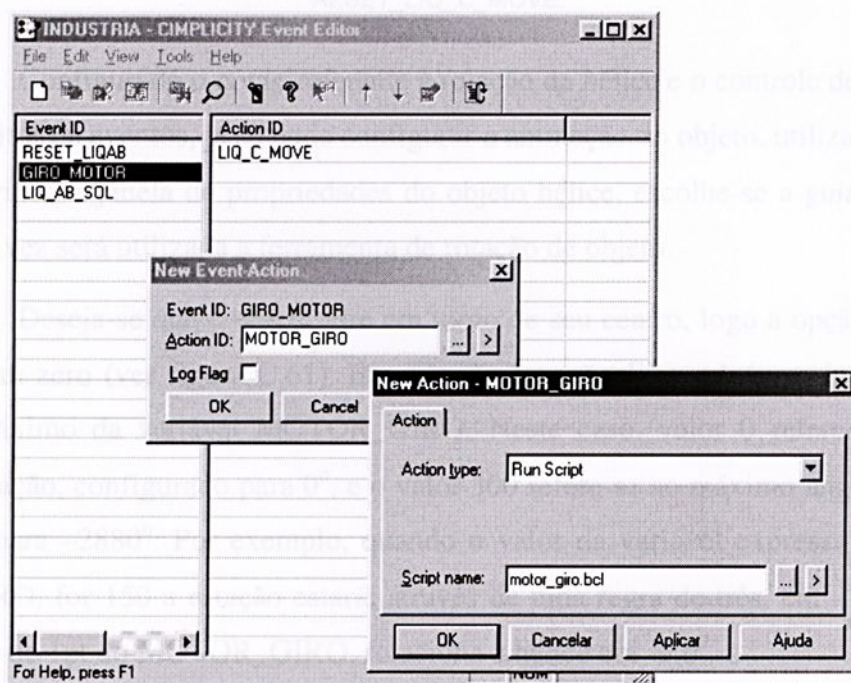
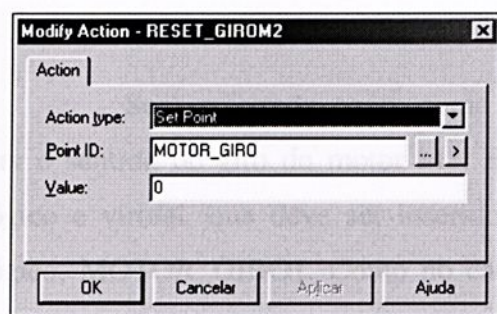


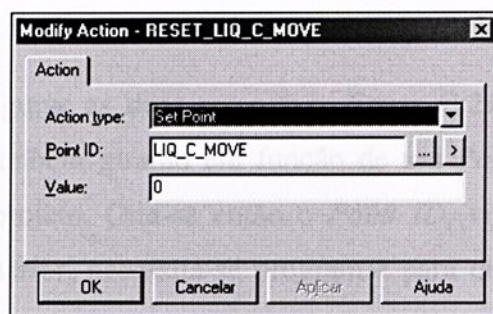
Figura C.59 - inserindo uma nova ação para o evento GIRO_MOTOR.

Assim, sempre que o *Point ID* Q06_PLC2 estiver em nível lógico “1”, ele executará as ações LIQ_C_MOVE e MOTOR_GIRO, que trata, respectivamente, do movimento do líquido do reservatório de mistura para o depósito C e da rotação da hélice do motor M2. Ou seja, essas duas ações ocorreram simultaneamente.

Os *Script* motor_giro.bcl e LIQ_C.bcl foram programados para finalizar quando o ponto Q06_PLC2 estiver em nível lógico “0”, contudo as variáveis e os *Point ID*, referentes a elas, continuarão com o valor da última contagem; assim deve-se voltar o valor de MOTOR_GIRO e LIQ_C_MOVE para zero sempre que o valor de Q06_PLC2 também for zero. Para isso, cria-se outro evento chamado RESET_M2_PLC2, que tem como ações zerar os *Point ID* MOTOR_GIRO e LIQ_C_MOVE, sempre que o valor Q06_PLC2 for zero, conforme mostra a figura C.60.



(a)



(b)

Figura C.60 - Novas ações para o novo evento RESET_M2_PLC2 – (a) ação RESET_GIROM2 (b) ação RESET_LIQ_C_MOVE.

Configurado o ponto referente à rotação da hélice e o controle de sua execução, através do editor de eventos, já se pode configurar a animação do objeto, utilizando o *software CimEdit*. Abrindo a janela de propriedades do objeto hélice, escolhe-se a guia *Rotation/Fill*, na qual desta vez será utilizada a ferramenta de rotação de objeto.

Deseja-se que o objeto gire em torno de seu centro, logo a opção *Center offset* é escolhida em zero (ver figura C.61). Em relação à opção *Exp. min/max*, insere-se o valor mínimo e máximo da variável MOTOR_GIRO. Neste caso, valor 0 refere-se ao mínimo ângulo de rotação, configurado para 0° , e o valor 300 refere-se ao máximo ângulo de rotação, configurado para -2880° . Por exemplo, quando o valor da variável expressa pelo *Point ID* MOTOR_GIRO, for 150 a rotação estará, através de uma regra de três, em -1440° , ou seja, cada variação de 1pt do MOTOR_GIRO, rotaciona a hélice em $-9,6^\circ$.

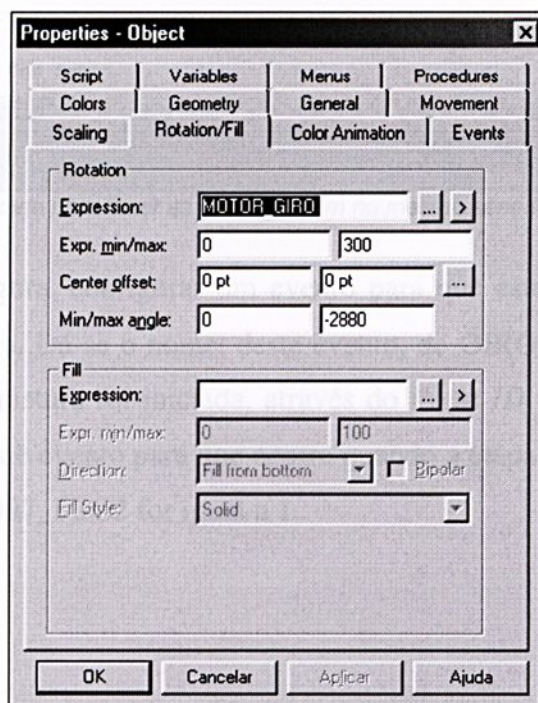


Figura C.61 - configurando a rotação da hélice do motor M2.

Será mostrado agora como configurar as duas setas (ver figura C.21) para indicar o sentido de giro do motor M1. Elas também girarão em função de um *Point ID*, analógico e virtual, que deve ser inserido no projeto. Cria-se então o *Point ID*, virtual e analógico, MOTOR_GIRO1. Como no caso anterior, necessita-se programar para que este ponto varie de 0 até 300.

Assim, escreve-se o programa, que estará em execução durante toda etapa de mistura. Salva-se este programa com o nome giro_motor1.bcl, conforme mostrado na figura C.62.

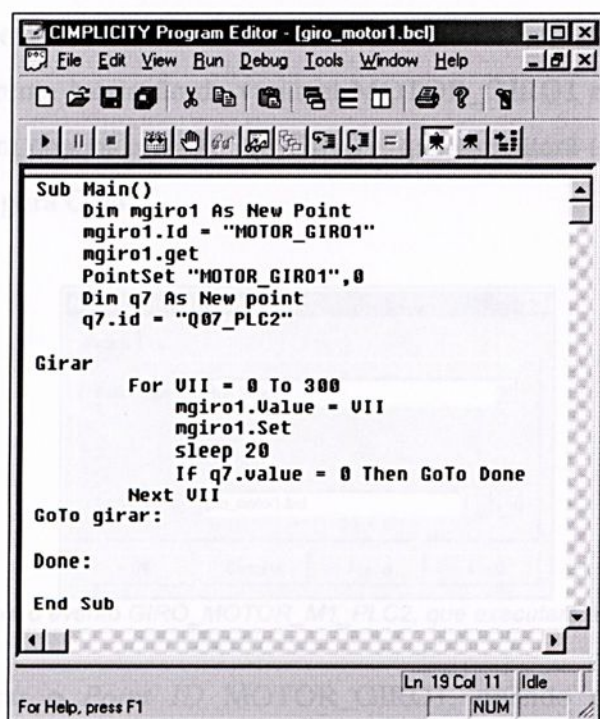


Figura C.62 - Script utilizado para fazer as setas girarem no mesmo sentido de rotação do motor M1.

Deve-se agora, configurar um evento para que este programa seja executado durante a etapa de mistura. Dá-se o nome, deste evento, de GIRO_MOTOR_M1_PLC2, que verificará se a etapa de mistura foi iniciada, através do *Point ID* Q07_PLC2. A figura C.63 mostra a configuração deste evento para que ocorra quando a etapa de mistura for iniciada, ou seja, quando o *Point ID* Q07_PLC2 for igual a 1.



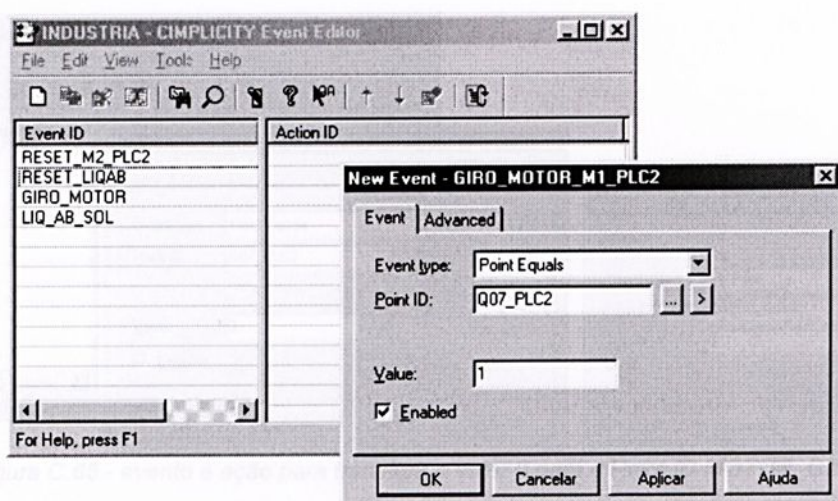


Figura C.63 - evento que verifica se a etapa de mistura está funcionando.

Caso a etapa de mistura tenha iniciado, o evento deverá efetuar a ação de executar o *Script*, referente à criação da variável MOTOR_GIRO1 e, para isso, configura-se uma nova ação, também chamada MOTOR_GIRO1, que executará o *Script* giro_motor1.bcl, conforme descrito na figura C.64.

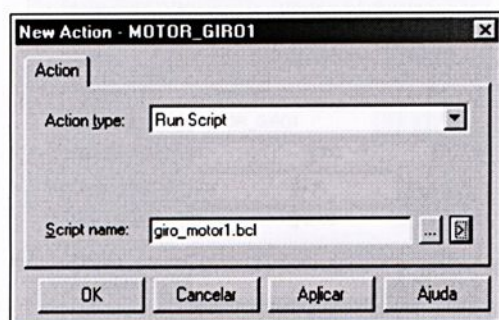


Figura C.64 - ação para o evento GIRO_MOTOR_M1_PL2, que executará a *Script* giro_motor1.bcl.

Para zerar o *Point ID* MOTOR_GIRO1, quando terminar o processo de mistura, projeta-se mais um evento com nome RESET_M1_PL2, cuja ação, criada com nome RESET_GIRO1, será de zerar o *Point ID* MOTOR_GIRO1, conforme apresentado na figura C.65.

Para que as setas fiquem visíveis somente durante o tempo em que o motor estiver ligado, lembrando que o motor M1 para antes de mudar o sentido de rotação, deve-se configurar sua visibilidade. Assim, configura-se também o item *Visibility* de sua *Color Animation*, para que o objeto fique visível somente quando estiver ligado em um determinado sentido de rotação, na configuração seta ou triângulo.

Para criar esta condição no item *Visibility* usa-se o comando (Q05_PL2 EQ 1 AND Q04_PL2 EQ 1) OR (Q05_PL2 EQ 1 AND Q03_PL2 EQ 1) que deixará a seta

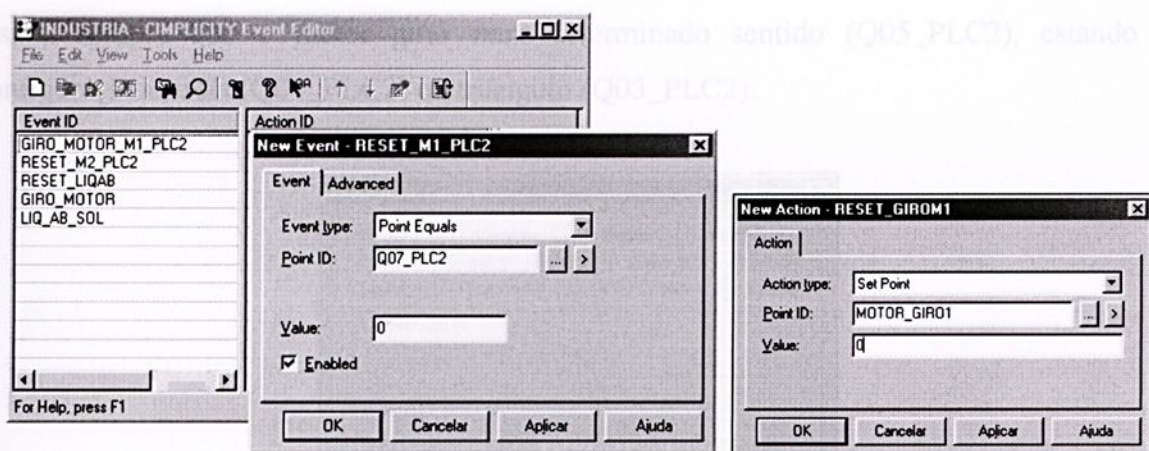


Figura C.65 - evento e ação para transferir o valor 0 para o Point ID MOTOR_GIRO1.

Pode-se agora configurar os objetos (setas circulares) no *CimEdit* para girarem, indicando o sentido de rotação do eixo do motor de mistura M1. Deve-se então abrir a janela de propriedades do objeto, neste caso uma das setas circulares, e configurar o *Rotation/Fill*, conforme mostra a figura C.66.

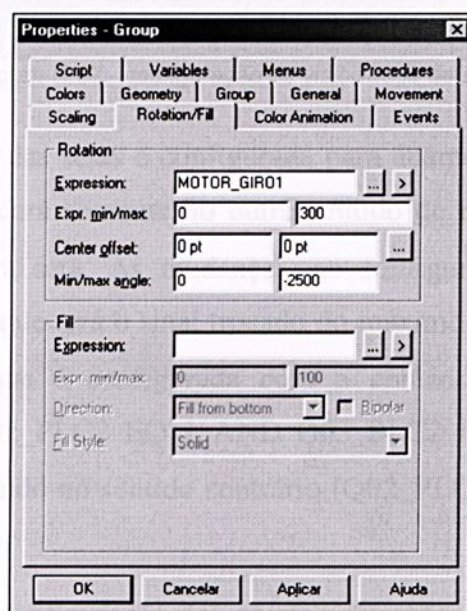


Figura C.66 - configuração de rotação da seta.

Para que as setas fiquem visíveis somente durante o tempo em que o motor estiver ligado, lembrando que o motor M1 pára antes de mudar o sentido de rotação, deve-se configurar sua visibilidade. Assim, configura-se também o item *Visibility* da guia *Color Animation*, para que o objeto fique visível somente quando estiver ligado em um determinado sentido de rotação, na configuração estrela ou triângulo.

Para criar esta condição no item *Visibility* usa-se o comando (Q05_PLC2 EQ 1 AND Q04_PLC2 EQ 1) OR (Q05_PLC2 EQ 1 AND Q03_PLC2 EQ 1), que deixará a seta

visível sempre que o motor girar num determinado sentido (Q05_PLC2), estando na configuração estrela (Q04_PLC2) ou triângulo (Q03_PLC2).

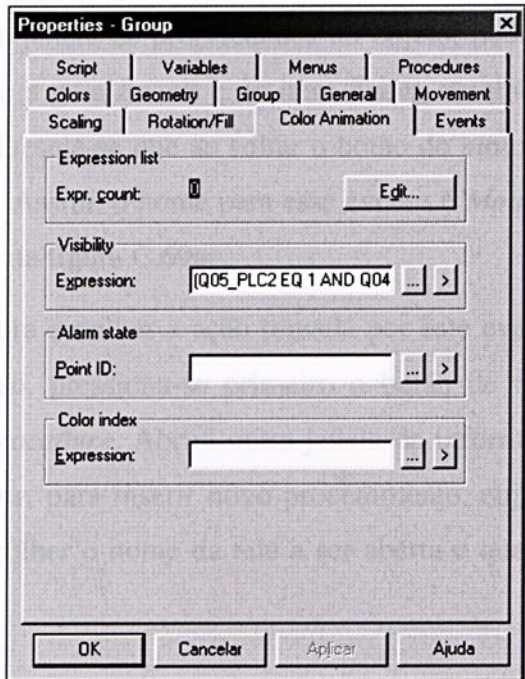


Figura C.67 - configuração de visibilidade da seta.

Assim, umas das setas é configurada para aparecer girando num determinado sentido. A outra seta aparecerá somente no outro sentido de rotação do motor, girando no sentido contrário a primeira seta. As mudanças, ao configurar a segunda seta, estão no máximo ângulo de rotação, que terá o sinal trocado do máximo ângulo da primeira seta, e na condição de visibilidade que é configurada com o comando (Q02_PLC2 EQ 1 AND Q04_PLC2 EQ 1) OR (Q02_PLC2 EQ 1 AND Q03_PLC2 EQ 1), ou seja, só aparecerá quando o motor estiver girando no sentido contrário (Q02_PLC2) à configurada na primeira seta.

Terminada a configuração dos objetos animados, falta agora apresentar como configurar um *Link* para abrir uma outra tela e como efetuar o controle dos CLPs de forma remota, utilizando o *Cimplicity HMI*.

Primeiro será mostrado como configurar um *Link*, tomando como exemplo o *Link* para a tela principal, ou tela geral, do projeto em questão. Primeiro escolhe-se o objeto que servirá como o caminho (*Link*) para abrir a nova tela. Neste caso é escolhido, através da biblioteca de objetos, o objeto da figura C.68.





Figura C.68 - objeto utilizado para funcionar como *Link* para outras telas.

Abrindo sua janela de propriedades do objeto, pode-se configura-lo como um *Link*. Primeiro seleciona-se a guia *Events* e no item *Event*, escolhe-se o evento com o qual a figura estará relacionada. Deseja-se que ao soltar o botão do *mouse* sobre o objeto, o mesmo direcione para a nova tela. Assim, o nome para esse evento é *Mouse Up*, selecionado no item *Event*, conforme mostrado na figura C.69a.

Deve-se agora escolher a ação tomada por este evento, que neste caso é abrir uma nova tela e, para isso, pressiona-se primeiro o botão de navegação do item *Action*, escolhendo o item *New Procedure*. Abrirá outra janela de informações do procedimento, na qual escolhe-se o item *New*, para inserir novo procedimento, cuja *Action type* será *Overlay screen*, restando agora escolher o nome da tela a ser aberta e qual o projeto base, conforme mostra a figura C.69b.

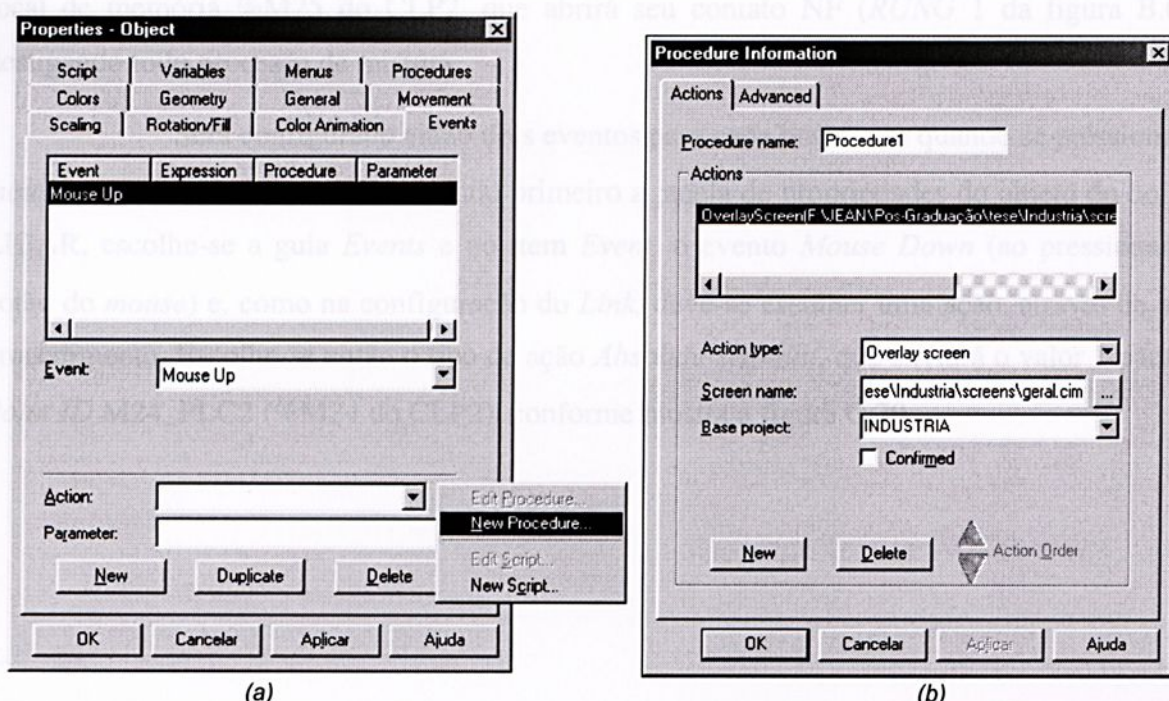


Figura C.69 - configurando o objeto *Link* – (a) abrindo novo procedimento ao soltarmos o botão esquerdo do mouse, (b) ação tomada ao soltar o botão esquerdo do mouse.

Desta forma, acabou-se de configurar o *Link* para a tela principal do projeto em questão. O outro *Link* presente na tela do misturador (ver figura C.21), refere-se à outra tela, contendo o gráfico de tendência da temperatura do líquido misturado, e este *Link* é

configurado da mesma forma, mudando-se apenas o caminho referente ao arquivo da tela a ser aberta.

Na tela do processo de mistura, restou-se apenas configurar a chave LIGA/DESLIGA o processo de mistura (ver figura C.21). Esta chave também é configurada utilizando a guia *Event* da janela de propriedades do objeto, contudo deve-se conhecer bem o processo e a forma como essa chave irá atuar no mesmo.

Os botões LIGA e DESLIGA são dois *Text Button* e deseja-se que eles funcionem como uma botoeira, ou seja, quando se pressionar com o botão esquerdo *mouse* ele envia um sinal de nível lógico “1” para a memória do CLP e quando o soltamos o sinal volta para o nível lógico “0”. Assim quando se pressionar o botão LIGA, este deverá enviar ao local de memória %M24, do CLP2, o sinal de nível lógico “1”, ligando o processo de mistura. Ao soltá-lo, o local de memória %M24 volta para “0”, contudo seu contato deverá ser selado na lógica contida no CLP2 (ver *RUNG 1* da figura B.6).

Quando se pressiona o botão DESLIGA, este envia nível lógico “1” para o local de memória %M25 do CLP2, que abrirá seu contato NF (*RUNG 1* da figura B.6), desligando todo processo de mistura.

Será configurado então dois eventos para cada botão, um quando se pressiona o *mouse* e outro quando se solta. Abrindo primeiro a janela de propriedades do objeto do botão LIGAR, escolhe-se a guia *Events* e no item *Event*, o evento *Mouse Down* (ao pressionar o botão do *mouse*) e, como na configuração do *Link*, deve-se escolher uma ação, através de um procedimento. Escolhe-se então o tipo de ação *Absolute setpoint*, que enviará o valor 1 para o *Point ID* M24_PLC2 (%M24 do CLP2), conforme mostra a figura C.70.

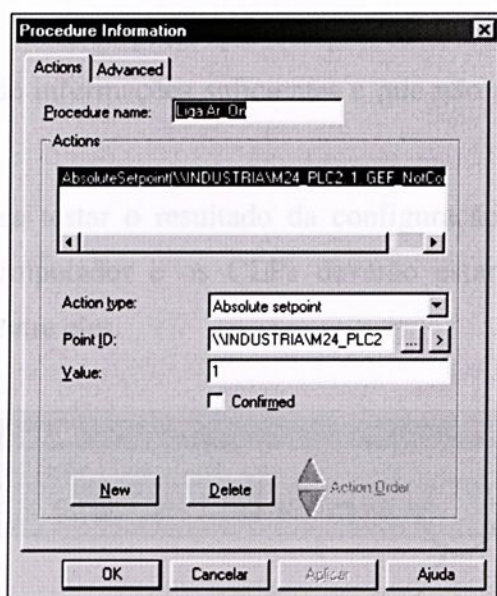


Figura C.70 - configurando o botão LIGAR para enviar "1" para o local de memória %M24 do CLP2, quando se pressiona o botão esquerdo do mouse.

De volta a guia *Events*, cria-se um novo evento, pressionando no botão *New*; este será o evento de soltar o *mouse*, ou seja, *Mouse Up*. A ação deste evento será enviar o nível lógico "0" para o *Point ID* M24_PLC2 ao se soltar o botão esquerdo do *mouse*, conforme figura C.71.

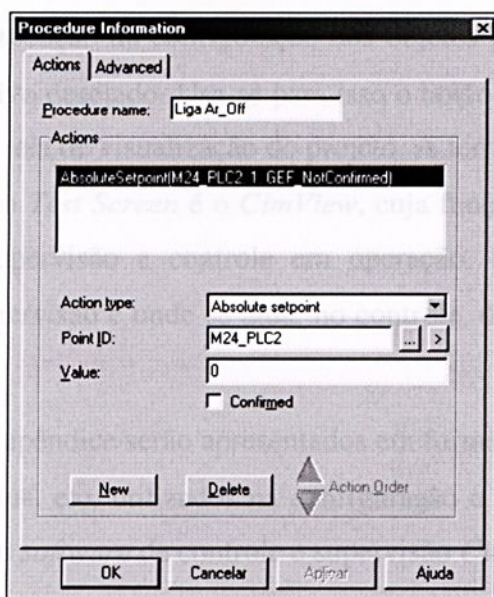


Figura C.71 - configurando o botão LIGAR para enviar "0" para o local de memória %M24 do CLP2, quando o botão esquerdo do mouse é solto.

A configuração do botão DESLIGAR é semelhante ao do botão ligar, mudando-se apenas o *Point ID* que será trabalhado. Assim conclui-se a configuração de todos os pontos animados da tela de supervisão e controle do processo de mistura, que é um dos segmentos da proposta da pequena indústria. É claro que pode-se incluir novos pontos, novas

animações e *Link*, porém deve-se preocupar sempre em manter, na tela de supervisão e controle, uma quantidade de informações suficientes e que não gere confusão na leitura dos dados.

Pode-se agora testar o resultado da configuração dos pontos desta parte do projeto e, para isso, o computador e os CLPs deverão estar ligados na rede e com a comunicação estabelecida entre eles.

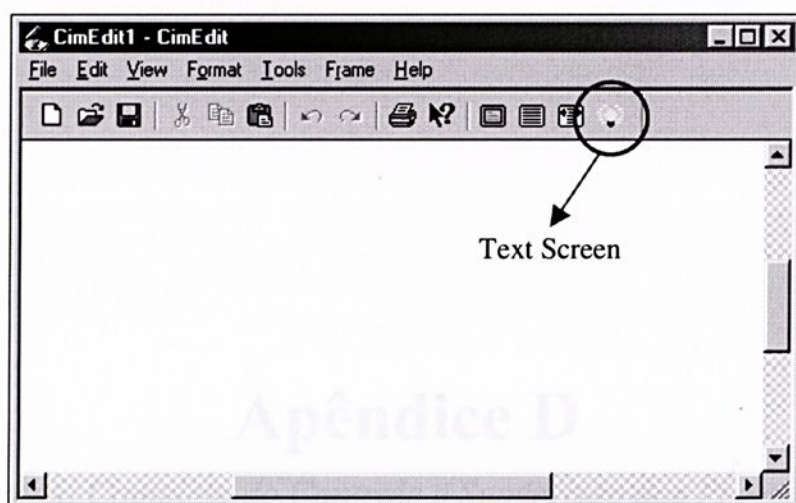


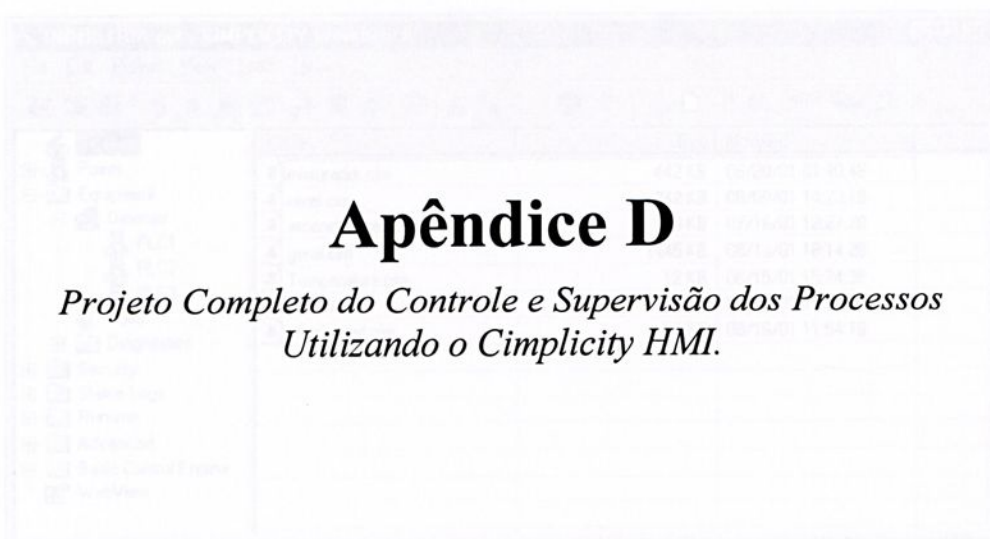
Figura C.72 - botão usado para abrir o software CimView.

Durante a execução da configuração dos objetos pode-se testar se o resultado está de acordo com o objetivo desejado. Usa-se para isso o botão *Test Screen*, que é o mesmo botão utilizado para abrir a tela de visualização do projeto. A ferramenta do *software* que abre quando se pressiona o botão *Test Screen* é o *CimView*, cuja função é única e exclusivamente para mostrar a tela de supervisão e controle em operação, onde se verifica os pontos configurados na tela de supervisão e onde se atua, no controle, de forma a ligar e desligar os processos remotamente.

No próximo apêndice serão apresentados em forma de figuras, todos os pontos, telas, programas, dispositivos, etc, utilizados na configuração e confecção da proposta final (todos segmentos), usando o *software* de controle e supervisão *Cimplicity-HMI*.

PROJETO COMPLETO DO CONTROLE E SUPERVISÃO DOS PROCESSOS UTILIZANDO O SIMPLICITY HMI.

D.1 SCREENS



Apêndice D

Projeto Completo do Controle e Supervisão dos Processos Utilizando o Simplicity HMI.

Figura D 1 - imagens referentes às telas do projeto feitas de acordo com o layout de projeto.

D.2 DEVICES

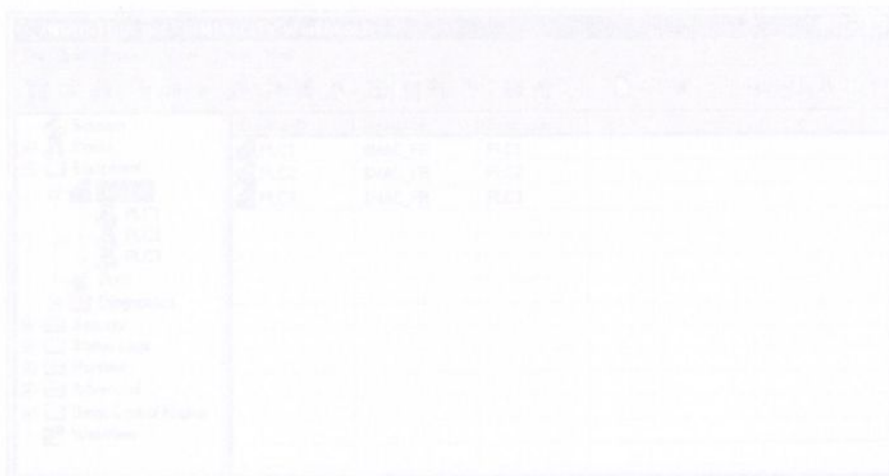


Figura D 2 - as CLPs configuradas no projeto.

APÊNDICE D

PROJETO COMPLETO DO CONTROLE E SUPERVISÃO DOS PROCESSOS UTILIZANDO O CIMPLICITY HMI.

D.1 SCREENS

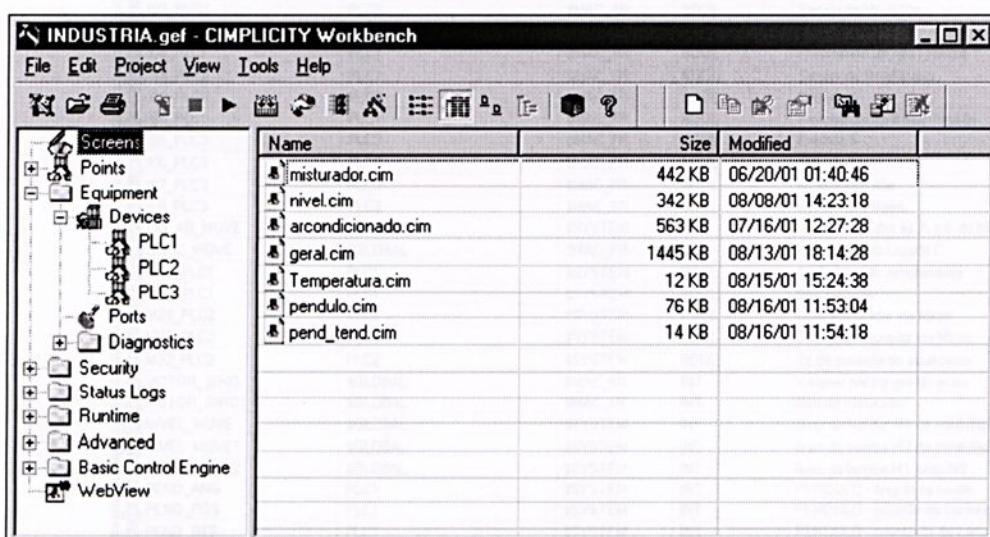


Figura D.1 - arquivos referentes às telas da proposta final de projeto para ser utilizado no laboratório de ensino.

D.2 DEVICES

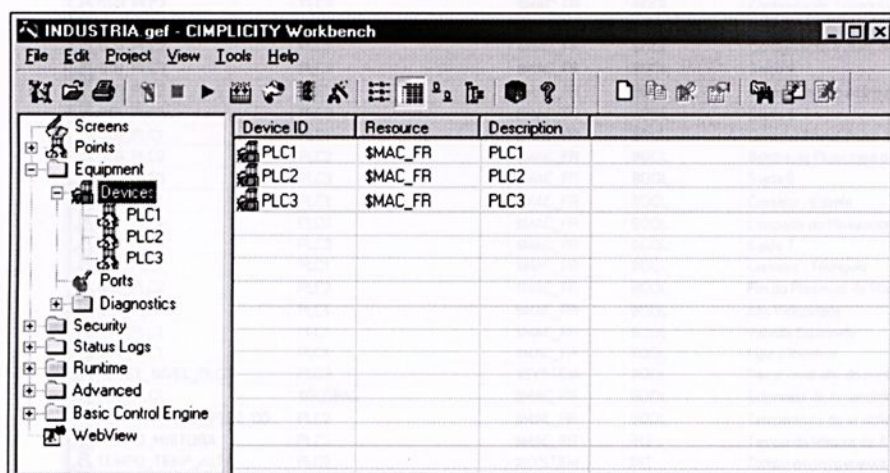


Figura D.2 - os CLPs configurados no projeto.

D.3 POINTS /ID

INDUSTRIA gel - CIMPLICITY Workbench					
File Edit Project View Tools Help					
Screens					
Point ID					
Equipment					
Security					
Status Logs					
Runtime					
Advanced					
Basic Control Engine					
WebView					
Point ID	Device ID	Resource	Point Type	Description	
AQ01_PLC1	PLC1	\$SYSTEM	INT	Sinal de referência p/ o pêndulo	
BOMBA_M1AR	\$GLOBAL	\$MAC_FR	INT	Giro da bomba M1 do ar	
BOMBA_M1NIVEL	\$GLOBAL	\$MAC_FR	INT	Giro da bomba M1 do nível	
BOMBA_M2AR	\$GLOBAL	\$MAC_FR	INT	Giro da bomba M2 do ar	
BOMBA_M2NIVEL	\$GLOBAL	\$MAC_FR	INT	Giro da bomba M2 do nível	
COND_MOVE	\$GLOBAL	\$MAC_FR	INT	Movimento - Água do condensador	
GAS_MOVEMENT	\$GLOBAL	\$MAC_FR	INT	Movimento do Gas	
GAS_MOVEMENT1	\$GLOBAL	\$MAC_FR	INT	Movimento do Gas depois da solenoide	
GELADA_MOVE	\$GLOBAL	\$MAC_FR	INT	Movimento - Água Gelada	
I01_PLC1	PLC1	\$SYSTEM	BOOL	Chave Geral	
I01_PLC2	PLC2	\$MAC_FR	BOOL	Botoeira Desliga	
I01_PLC3	PLC3	\$MAC_FR	BOOL	Entrada 1	
I02_PLC1	PLC1	\$MAC_FR	BOOL	Termostato - Condensador	
I02_PLC2	PLC2	\$MAC_FR	BOOL	Botoeira Liga	
I02_PLC3	PLC3	\$MAC_FR	BOOL	Entrada 2	
I03_PLC1	PLC1	\$MAC_FR	BOOL	Termostato - Água Gelada	
I03_PLC2	PLC2	\$MAC_FR	BOOL	Sensor de Nível Alto	
I03_PLC3	PLC3	\$MAC_FR	BOOL	Entrada 3	
I04_PLC1	PLC1	\$MAC_FR	BOOL	Pressostato de Alta Pressão	
I04_PLC2	PLC2	\$MAC_FR	BOOL	Sensor de Nível Baixo	
I04_PLC3	PLC3	\$MAC_FR	BOOL	Entrada 4	
I05_PLC1	PLC1	\$MAC_FR	BOOL	Pressostato de Baixa Pressão	
I05_PLC3	PLC3	\$MAC_FR	BOOL	Entrada 5	
I06_PLC3	PLC3	\$MAC_FR	BOOL	Entrada 6	
I07_PLC3	PLC3	\$MAC_FR	BOOL	Nível Muito Alto	
I08_PLC3	PLC3	\$MAC_FR	BOOL	Nível Muito Baixo	
LIQ_AB_MOVE	\$GLOBAL	\$SYSTEM	INT	Movimentos dos liq. A e B do Misturador	
LIQ_C_MOVE	\$GLOBAL	\$MAC_FR	INT	Movimento do Líquido C	
M10_PLC1	PLC1	\$SYSTEM	BOOL	Botão Liga/Ar remotamente	
M15_PLC1	PLC1	\$SYSTEM	BOOL	ligar pendulo	
M24_PLC2	PLC2	\$SYSTEM	BOOL	Ligar Misturador via Micro	
M25_PLC2	PLC2	\$SYSTEM	BOOL	Desligar Misturador via Micro	
M32_PLC2	PLC2	\$SYSTEM	BOOL	Bit de controle do aquecedor	
MOTOR_GIRO	\$GLOBAL	\$MAC_FR	INT	Variável para o giro do motor	
MOTOR_GIRO1	\$GLOBAL	\$MAC_FR	INT	Giro do misturador	
NIVEL_MOVE	\$GLOBAL	\$SYSTEM	INT	fluxo da bomba M1 do controlador de nive	
NIVEL_MOVE1	\$GLOBAL	\$SYSTEM	INT	fluxo da bomba M2 do controlador de nive	
NIVEL_MOVE2	\$GLOBAL	\$SYSTEM	INT	fluxo da bomba M1 e/ou M2	
PEND_ANG	PLC1	\$SYSTEM	INT	PENDULO - ângulo da haste	
PEND_POS	PLC1	\$SYSTEM	INT	PENDULO - posição do carrinho	
PEND_REF	PLC1	\$SYSTEM	INT	PENDULO - referência do carro	
PID_CV_AQ01	PLC2	\$SYSTEM	INT	Sinal de Controle _ DEGEM	
PID_FV_AI01	PLC2	\$SYSTEM	INT	Entrada de Temperatura _ DEGEM	
PID_SP_R450	PLC2	\$SYSTEM	INT	Sinal de Referência _ DEGEM	
Q01_PLC1	PLC1	\$MAC_FR	BOOL	Bomba de água gelada	
Q01_PLC2	PLC2	\$MAC_FR	BOOL	Valvulas Solenoides	
Q01_PLC3	PLC3	\$MAC_FR	BOOL	Saida 1	
Q02_PLC1	PLC1	\$MAC_FR	BOOL	Bomba de água do Condensador	
Q02_PLC2	PLC2	\$MAC_FR	BOOL	Sentido de Giro Horário	
Q02_PLC3	PLC3	\$MAC_FR	BOOL	Saida 2	
Q03_PLC2	PLC2	\$MAC_FR	BOOL	Configuração Triângulo do Misturador	
Q03_PLC3	PLC3	\$MAC_FR	BOOL	Saida 3	
Q04_PLC2	PLC2	\$MAC_FR	BOOL	Configuração Estrela do Misturador	
Q04_PLC3	PLC3	\$MAC_FR	BOOL	Saida 4	
Q05_PLC2	PLC2	\$MAC_FR	BOOL	Sentido de Giro Anti-horário	
Q05_PLC3	PLC3	\$MAC_FR	BOOL	Saida 5	
Q06_PLC1	PLC1	\$MAC_FR	BOOL	Contator Ramo direto	
Q06_PLC2	PLC2	\$MAC_FR	BOOL	Bomba de Fluxo para o Depósito	
Q06_PLC3	PLC3	\$MAC_FR	BOOL	Saida 6	
Q07_PLC1	PLC1	\$MAC_FR	BOOL	Contator - Estrela	
Q07_PLC2	PLC2	\$MAC_FR	BOOL	Lampada do Misturador	
Q07_PLC3	PLC3	\$MAC_FR	BOOL	Saida 7	
Q08_PLC1	PLC1	\$MAC_FR	BOOL	Contator - Triângulo	
Q08_PLC2	PLC2	\$MAC_FR	BOOL	Fim do Processo de Mistura	
Q09_PLC1	PLC1	\$MAC_FR	BOOL	Alta Velocidade	
Q11_PLC1	PLC1	\$MAC_FR	BOOL	Valvula Solenoide	
Q12_PLC1	PLC1	\$MAC_FR	BOOL	Liga o Inversor	
RESET_NIVEL_PLC3	PLC3	\$SYSTEM	BOOL	Reset nível alto do controlador de nível	
SOL_PLC1	\$GLOBAL	\$MAC_FR	BOOL	Solenóide do ArCondicionado	
TEMPERATURA_PLC2_CO...	PLC2	\$MAC_FR	BOOL	Temperatura do ar obtida pela CommReq	
TEMPO_MISTURA	PLC2	\$MAC_FR	INT	Tempo da Mistura de A e B	
TEMPO_TEMP_ALTA	PLC2	\$SYSTEM	INT	Tempo em temperatura alta	
TEMPO_VENTILADOR	PLC1	\$SYSTEM	INT	Tempo para o ventilador entrar em alta	

Figura D.3 - Point ID configurados durante execução do projeto.

D.4 PORTS

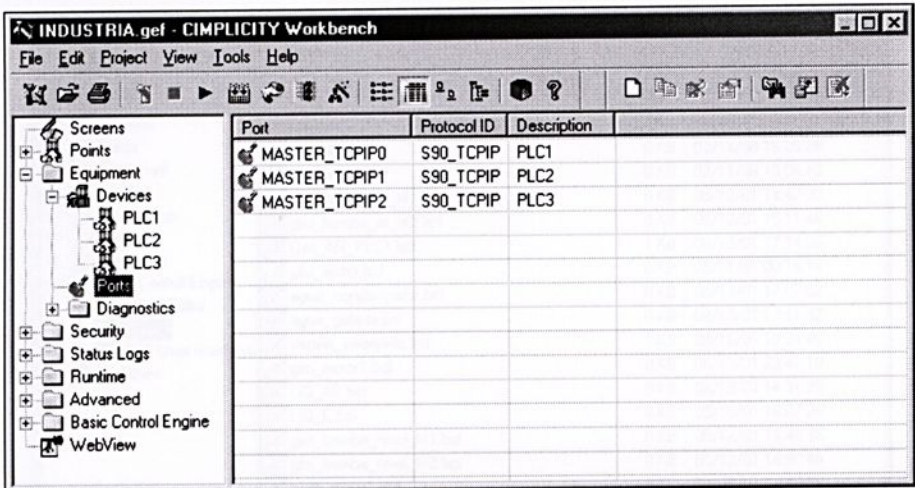


Figura D.4 - portas de comunicação utilizadas entre o software de supervisão e controle e os CLPs.

D.5 EVENTS ID E ACTION ID



Figura D.5 - eventos e ações configurados para aturarem em todas as telas.



D.6 SCRIPTS

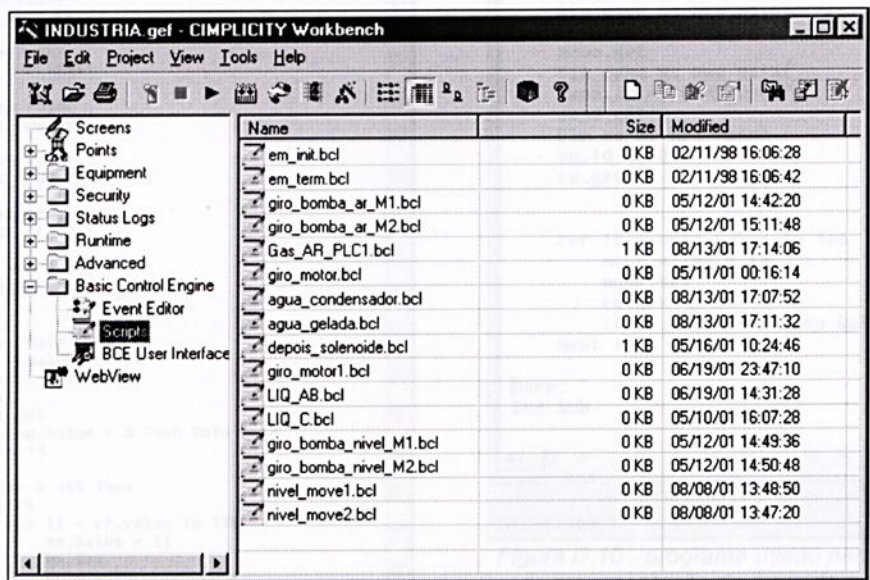


Figura D.6 - arquivos referentes aos programas utilizados no projeto.

Apresenta-se a seguir, os programas (*Scripts*) utilizados para criar as variações dos pontos virtuais, que geram, na tela, animações dinâmicas como fluxo de líquido, movimento do motor, etc.

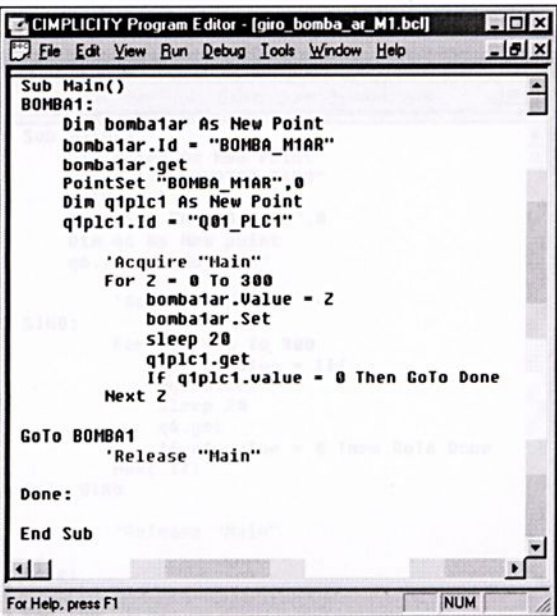


Figura D.7 - programa usado para criar a variável de rotação da hélice do motor M1, do sistema de refrigeração.

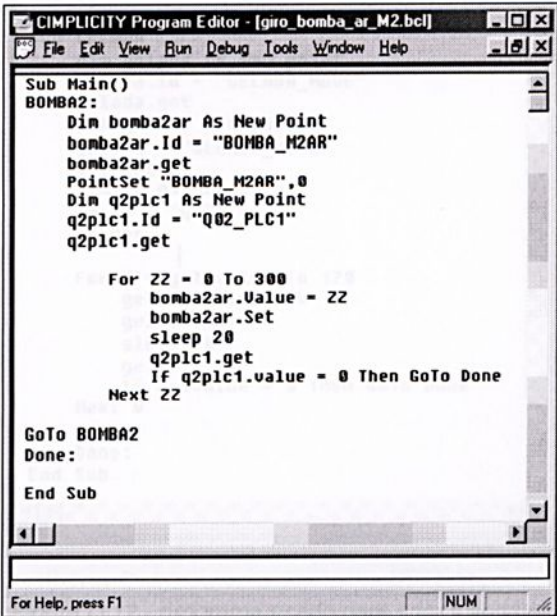


Figura D.8 - programa usado para criar a variável de rotação da hélice do motor M2, do sistema de refrigeração.




```

CIMPLICITY Program Editor - [gas_AR_PLC1.bcl]
File Edit View Run Debug Tools Window Help

Sub Main()
  Dim r As New Point
  r.Id = "GAS_MOVEMENT"
  r.get
  Dim x As New Point
  x.Id = "GAS_MOVEMENT"
  x.get
  Dim rr As New Point
  rr.Id = "GAS_MOVEMENT1"
  rr.get
  Dim xx As New Point
  xx.Id = "GAS_MOVEMENT1"
  xx.get
  Dim sol As New Point
  sol.Id = "SOL_PLC1"
  sol.get
  Dim comp As New Point
  comp.Id = "Q06_PLC1"
  comp.get

  'Acquire "Main"
  For I = r.value To 150
    x.Value = I
    x.set
    comp.get
    If comp.value = 0 Then GoTo Done
    Sleep 10
  Next I
  If x.value > 140 Then
    sol.get
    For II = rr.value To 105
      xx.Value = II
      xx.set
      comp.get
      If comp.value = 0 Then GoTo Done
      Sleep 10
    Next II
  End If
  'Release "Main"
Done:
End Sub

```

Figura D.9 - programa usado para criar a variável de movimentação do gás nos dutos do sistema de refrigeração, até a válvula solenóide.

```

CIMPLICITY Program Editor - [agua_condensador.bcl]
File Edit View Run Debug Tools Window Help

Sub Main()
  Dim move As New point
  move.Id = "COND_MOVE"
  move.get
  Dim cond As New point
  cond.Id = "COND_MOVE"
  cond.get
  Dim co As New Point
  co.Id = "Q02_PLC1"
  co.get

  For IU = cond.value To 166
    move.value = IU
    move.set
    co.get
    If co.value = 0 Then GoTo Done
  Next IU

Done:
End Sub

```

Figura D.10 - programa usado para criar a variável de movimentação da água do condensador do sistema de refrigeração.

```

CIMPLICITY Program Editor - [giro_motor.bcl]
File Edit View Run Debug Tools Window Help

Sub Main()
  Dim mgiro As New Point
  mgiro.Id = "MOTOR_GIRO"
  mgiro.get
  PointSet "MOTOR_GIRO",0
  Dim q6 As New point
  q6.Id = "Q06_PLC2"

  'Acquire "Main"
GIRO:
  For III = 0 To 300
    mgiro.Value = III
    mgiro.set
    sleep 20
    q6.get
    If q6.value = 0 Then GoTo Done
  Next III
GoTo GIRO

  'Release "Main"
Done:
End Sub

```

Figura D.11 - programa usado para criar a variável de rotação da hélice do motor M2 do processo de mistura e aquecimento de líquido.

```

CIMPLICITY Program Editor - [agua_gelada.bcl]
File Edit View Run Debug Tools Window Help

Sub Main()
  GELADA
End Sub

Sub GELADA
  Dim gelada As New point
  gelada.Id = "GELADA_MOVE"
  gelada.get
  Dim gel As New point
  gel.Id = "GELADA_MOVE"
  gel.get
  Dim ge As New point
  ge.Id = "Q02_PLC1"
  ge.get

  For U = gel.value To 170
    gelada.value = U
    gelada.set
    sleep 20
    ge.get
    If ge.value = 0 Then GoTo Done
  Next U

Done:
End Sub

```

Figura D.12 - programa usado para criar a variável de movimentação da água gelada do sistema de refrigeração.

```

CIMPLICITY Program Editor - [depois_solenóide.bcl]
File Edit View Run Debug Tools Window Help

Sub Main()
  Dim x As New point
  x.id = "GAS_MOVEMENT"
  x.get
  Dim xx As New point
  xx.id = "GAS_MOVEMENT1"
  xx.get
  Dim rr As New point
  rr.id = "GAS_MOVEMENT1"
  rr.get
  Dim sol As New point
  sol.id = "SOL_PL1"
  Dim comp As New Point
  comp.id = "Q06_PL1"
  comp.get

  If x.value > 149 Then
    sol.get
    For UI = rr.value To 105
      xx.value = UI
      xx.set
      comp.get
      If comp.value = 0 Then GoTo Done
      Sleep 10
    Next UI
  End If

Done:
End Sub

```

Figura D.13 - programa usado para criar a variável de movimentação do gás após a válvula solenóide do sistema de refrigeração.

```

CIMPLICITY Program Editor - [giro_motor1.bcl]
File Edit View Run Debug Tools Window Help

Sub Main()
  Dim mgiro1 As New Point
  mgiro1.id = "MOTOR_GIRO1"
  mgiro1.get
  PointSet "MOTOR_GIRO1", 0
  Dim q7 As New point
  q7.id = "Q07_PL1"

  Girar
    For VII = 0 To 300
      mgiro1.value = VII
      mgiro1.set
      sleep 20
      If q7.value = 0 Then GoTo Done
    Next VII
  GoTo Girar:

Done:
End Sub

```

Figura D.15 - programa usado para criar a variável de rotação das setas do processo de mistura e aquecimento de líquido.

```

CIMPLICITY Program Editor - [LIQ_AB.bcl]
File Edit View Run Debug Tools Window Help

Sub Main()
  Dim liqab As New Point
  liqab.id = "LIQ_AB_MOVE"
  liqab.get
  Dim liquab As New Point
  liquab.id = "LIQ_AB_MOVE"
  liquab.get
  Dim comp As New Point
  comp.id = "Q01_PL1"

  For VIII = liqab.value To 63
    liquab.value = VIII
    liquab.set
    comp.get
    If comp.value = 0 Then GoTo Done
    Sleep 10
  Next VIII

Done:
End Sub

```

Figura D.14 - programa usado para criar a variável de movimentação dos líquidos A e B do processo de mistura e aquecimento de líquidos.

```

CIMPLICITY Program Editor - [LIQ_C.bcl]
File Edit View Run Debug Tools Window Help

Sub Main()
  Dim liqc As New Point
  liqc.id = "LIQ_C_MOVE"
  liqc.get
  Dim liquc As New Point
  liquc.id = "LIQ_C_MOVE"
  liquc.get

  For IX = liqc.value To 83
    liquc.value = IX
    liquc.set
    Sleep 10
  Next IX

Done:
End Sub

```

Figura D.16 - programa usado para criar a variável de movimentação do líquido misturado para o depósito C, no processo de mistura e aquecimento de líquidos.


```

Sub Main()
BOMBA1:
Dim bomba1nivel As New Point
bomba1nivel.Id = "BOMBA_M1NIVEL"
bomba1nivel.get
PointSet "BOMBA_M1NIVEL",0

Dim q2plc3 As New Point
q2plc3.Id = "Q02_PLC3"
q2plc3.get

For S = 0 To 300
bomba1nivel.Value = S
bomba1nivel.Set
sleep 20
q2plc3.get
If q2plc3.value = 0 Then GoTo Done

Next S

GoTo BOMBA1

Done:
End Sub

```

Ln 25 Col 0 Idle
For Help, press F1

Figura D.17 - programa usado para criar a variável de rotação da hélice do motor M1 do controlador de nível.

```

Sub Main()
BOMBA2:
Dim bomba2nivel As New Point
bomba2nivel.Id = "BOMBA_M2NIVEL"
bomba2nivel.get
PointSet "BOMBA_M2NIVEL",0

Dim q1plc3 As New Point
q1plc3.Id = "Q01_PLC3"
q1plc3.get

For SS = 0 To 300
bomba2nivel.Value = SS
bomba2nivel.Set
sleep 20
q1plc3.get
If q1plc3.value = 0 Then GoTo Done

Next SS

GoTo BOMBA2

Done:
End Sub

```

Ln 22 Col 0 Idle
For Help, press F1

Figura D.18 - programa usado para criar a variável de rotação da hélice do motor M2 do controlador de nível.

```

Sub Main()
Dim nivel As New point
nivel.Id = "NIVEL_MOVE"
nivel.get
Dim niv As New point
niv.Id = "NIVEL_MOVE"
niv.get
Dim niv3 As New point
niv3.Id = "NIVEL_MOVE2"

For IV = niv.value To 100
nivel.value = IV
nivel.set
Next IV

niv3.get
If niv3.value = 0 Then
For I = 0 To 55
niv3.value = I
niv3.set
Next I
End If

End Sub

```

Ln 25 Col 0 Idle
For Help, press F1

Figura D.19 - programa usado para criar a variável de movimentação do líquido do reservatório para o depósito A ou B, através do motor M1, do controlador de nível.

```

Sub Main()
Dim nivel As New point
nivel.Id = "NIVEL_MOVE1"
nivel.get
Dim niv As New point
niv.Id = "NIVEL_MOVE1"
niv.get
Dim niv3 As New point
niv3.Id = "NIVEL_MOVE2"

For IV = niv.value To 100
nivel.value = IV
nivel.set
Next IV

niv3.get
If niv3.value = 0 Then
For I = 0 To 55
niv3.value = I
niv3.set
Next I
End If

End Sub

```

Ln 23 Col 0 Idle
For Help, press F1

Figura D.20 - programa usado para criar a variável de movimentação do líquido do reservatório para o depósito A ou B, através do motor M2, do controlador de nível.



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Câmpus de Ilha Solteira
Programa de Pós-Graduação em Engenharia Elétrica
Av. Brasil Centro, 56
15385-000 Ilha Solteira - SP
www.dee.feis.unesp.br

