

TESE DEFENDIDA POR Maurício Pires
Martins E APROVADA PELA
COMISSÃO JULGADORA EM 21/12/98

João Maurício Rosário
ORIENTADOR

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO

Estruturação da Parte de Comando de um Sistema Automatizado de Produção com Ênfase na Implementação de um Sistema de Supervisão

Autor : **Maurício Pires Martins**

Orientador: **João Maurício Rosário**

Curso: Engenharia Mecânica.

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Tese de doutorado apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para obtenção do título de Doutor em Engenharia Mecânica.

Campinas, 1998

S.P. – Brasil

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO**

**Estruturação da Parte de Comando de um Sistema
Automatizado de Produção com Ênfase na
Implementação de um Sistema de Supervisão**

Autor : Mauricio Pires Martins

Orientador: João Maurício Rosário

Curso: Engenharia Mecânica.

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Tese de doutorado apresentada à comissão de Pós Graduação da Faculdade de Engenharia Mecânica, como requisito para obtenção do título de Doutor em Engenharia Mecânica.

Campinas, 1998

S.P. – Brasil

7916036



UNIVERSIDADE	BC
N.º DE INSCRIÇÃO	
V.º DE INSCRIÇÃO	
V.º DE INSCRIÇÃO	
T.º DE INSCRIÇÃO	38467
DATA	229/99
PRECIO	R\$ 11,00
DATA	04/08/99
N.º CPD	

CM-00125613-9

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

M366e

Martins, Mauricio Pires

Estruturação da parte de comando de um sistema automatizado de produção com ênfase na implementação de um sistema de supervisão. / Mauricio Pires Martins.-- Campinas, SP: [s.n.], 1998.

Orientador: João Maurício Rosário.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Automação industrial - Protocolos. 2. Controladores programáveis. 3. Sistemas de tempo discreto. 4. Controle de produção - Planejamento. I. Rosário, João Maurício. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

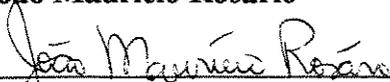
**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO**

TESE DE DOUTORADO

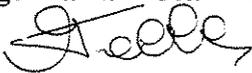
**Estruturação da Parte de Comando de um Sistema
Automatizado de Produção com Ênfase na
Implementação de um Sistema de Supervisão**

Autor : **Mauricio Pires Martins**

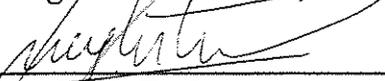
Orientador: **João Maurício Rosário**



**Prof. Dr. João Maurício Rosário, Presidente
Faculdade de Engenharia Mecânica, UNICAMP**



**Prof. Dr. Geraldo Nonato Telles
Faculdade de Engenharia Mecânica, UNICAMP**



**Prof. Dr. Luis Gonzaga Trabasso
Instituto Tecnológico da Aeronáutica**



**Prof. Dr. Franco Giuseppe Dedini
Faculdade de Engenharia Mecânica, UNICAMP**



**Prof. Dr. Glauco Augusto de Paula Caurin
Engenharia Mecânica, USP - São Carlos
Universidade de Mogi das Cruzes**

Campinas, 21 de Dezembro de 1998

Dedicatória

Dedico este trabalho à minha esposa Íris, por seu amor e paciência, e aos meus pais, pelo apoio e incentivo.

Agradecimentos

Este trabalho não poderia ser terminado sem ajuda de diversas pessoas às quais presto minha homenagem:

Ao meu orientador, prof. Dr. João Maurício Rosário, pela orientação e suporte no desenvolvimento deste trabalho.

Ao amigo Jorge Giles Ferrer pelo incentivo e amizade.

Ao colega Almiro Franco da Silveira Júnior e a todos os colegas do Departamento de Projeto Mecânico que ajudaram de forma direta ou indireta para a conclusão deste trabalho.

Ao Serviço Nacional de Aprendizagem Industrial – SENAI pelo apoio à realização deste trabalho.

À CAPES-COFECUB, que pelo auxílio ao projeto (nº 195/96) de cooperação bilateral Brasil-França, propiciou a construção da Plataforma PIPEFA.

Resumo

MARTINS, Mauricio Pires. *Estruturação da Parte de Comando de um Sistema Automatizado de Produção com Ênfase na Implementação de um Sistema de Supervisão*. Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1998. 128 p. Tese (Doutorado).

Este trabalho apresenta um estudo da automação relativo ao planejamento do processo, monitoração de máquinas e equipamentos, fluxo de materiais, supervisão do sistema e redes de comunicação. No trabalho são desenvolvidos programas para monitoração de máquinas e equipamentos. São apresentados e implementados algoritmos e soluções para o uso de interfaces, utilizando-se como hardware básico, um microcomputador e como linguagem de programação, o Visual Basic. São apresentados exemplos de sistemas automatizados desenvolvidos e implementados no Laboratório de Automação, onde são feitos testes comparativos de programação em Diagrama de Relés e Grafcet. Baseado na hierarquia do planejamento de sistemas CIM (Computer Integrated Manufacturing), é implementado um sistema de supervisão ao ambiente experimental PIPEFA (Plataforma Industrial para Pesquisa Ensino e Formação em Automação), que simula um sistema integrado de produção. Utilizando-se de redes e interfaces de comunicação, o sistema de supervisão integra os diversos níveis da pirâmide CIM, fazendo a comunicação do nível administrativo com o de produção.

Palavras-Chave:

- Automação, Programação de CLP, Grafcet, Sistemas Discretos, Planejamento e Controle da Produção.

Abstract

MARTINS, Mauricio Pires. *Command Part Structuration of Manufacturing Automation System with Emphasis in Supervisory System*. Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1998. 128. p. Tese (Doutorado).

This work presents a study of automation relative to process planning, monitoring of machines and equipment, material flux, supervision of the system and communication network. In this work are developed programs to monitoring machines and equipments. Algorithms and solutions to the use of interfaces are presented and implemented, utilizing a microcomputer as basic hardware and Visual Basic as the programming language. Examples of automated systems are presented. They are developed and implemented in the automation laboratory “Laboratório de Automação Integrada e Robótica”, where comparative tests of programming in Ladder Diagram and Grafset are made. Based on the hierarchy of planning of the CIM systems, it is implemented a supervisory system to the experimental ambient PIPEFA (Industrial Platform to the Research, Education and Formation in Automation), that simulates a integrated system of production. Utilized by networks and interfaces of communication, the supervisory system integrates different levels of the CIM pyramid, making a communication of the administrative level with the production one, by Internet.

Key Words:

- Automation, PLC programming, Grafset, Discrete Systems, Production Planning and Control

Índice

Capítulo 1 Introdução.....	1
1.1 Objetivos.....	4
1.2 Delineamento do Trabalho	4
Capítulo 2 Revisão Bibliográfica.....	6
2.1 Conceitos Básicos	6
2.2 Descrição de Sistemas Seqüenciais.....	12
2.3 Controlador Lógico Programável.....	14
2.4 Diagrama de Relés versus Diagrama Seqüencial.....	16
2.5 Automação da Manufatura	17
2.6 Sistema de Supervisão.....	21
2.7 Ambiente de Programação.....	26
Capítulo 3 Descrição do Projeto PIPEFA.....	28
3.1 Funcionamento do Sistema.....	30
3.1.1 Posto de Montagem/Desmontagem de Produtos.....	33
3.1.2 Posto de Carregamento de Placas de Base.....	34
3.1.3 Posto de Descarregamento de Produtos.....	34
3.1.4 Posto de Inspeção	34
3.1.5 Sistema de Transferência	36
3.1.6 Leitura do Código de um Produto	36

3.1.7	Manipulador Robótico	36
3.1.8	Sistema de Supervisão	36
3.2	Descrição da Comunicação entre os Postos de Trabalho	38
3.3	Sistema de Gestão de Produção	45
Capítulo 4	Utilização de Sistemas de Supervisão em Automação	47
4.1	Sistema de Supervisão	47
4.2	Implementação de um Programa de Controle para Transporte de Peças	49
4.2.1	Programação do Transporte de Peças em Diagrama de Relés	50
4.2.2	Programação do transporte de peças em Grafcet	59
4.3	Programa de Controle para Estação de Montagem/Desmontagem.....	62
4.4	Implementação de um Programa de Monitoração para Centro de Usinagem	68
Capítulo 5	Implementação do Sistema de Supervisão ao Projeto PIPEFA	76
5.1	Programação do CLP Ligado ao Sistema de Supervisão	90
5.2	Teste do Sistema de Supervisão.....	94
5.3	Comparação com Sistema de Supervisão Industrial	96
Capítulo 6	Conclusões e Perspectivas Futuras.....	101
7.	Referências Bibliográficas.....	103
8.	Anexos.....	107
	Anexo I – Possíveis Configurações de Montagem.....	107
	Anexo II – Relação das Variáveis de Entrada/Saída da Plataforma PIPEFA.....	108
	Anexo III – Grafcet do Sistema de Supervisão para o PIPEFA.....	110
	Anexo IV – Programa de Supervisão para o PIPEFA.....	114
	Anexo V - Interfaces de Comunicação	143
	Interface Paralela.....	143
	Interface de Comunicação Koyo.....	146

Anexo VI – Rotinas de Apoio	149
Programa para Aquisição de Dados da Interface Paralela.....	149
Aquisição de Dados do CLP da Koyo.....	151
Anexo VII – Programa do Transporte de Peças em Diagrama de Relés	154
Anexo VIII – Revisão de Portas Lógicas.....	158

Lista de Figuras

Figura 2.1: Arquitetura Genérica para Sistemas CIM (Teicholz,1989).....	8
Figura 2.2: Hierarquia do Sistema CIM.....	9
Figura 2.3: Parte de Comando e Operativa de um Elevador.....	14
Figura 2.4: Exemplo de SFC	16
Figura 2.5: Layout para Rede de Comunicação de CLPs	24
Figura 3.1: Placa Base e Placa de Montagem.....	30
Figura 3.2: Layout da Plataforma PIPEFA	33
Figura 3.3: Hierarquia de Funcionamento do Projeto PIPEFA.....	38
Figura 3.4: Representação esquemática do papel do sistema de Supervisão	39
Figura 3.5: Informações para o Sistema de Transferência.....	41
Figura 3.6: Informações para o Posto de Montagem/Desmontagem Central.....	41
Figura 3.7: Informações para o Posto de Montagem/Desmontagem Lateral.....	42
Figura 3.8: Informações do Posto de Controle de Qualidade.....	43
Figura 3.9: Informação para o Posto de Carregamento/Descarregamento	45
Figura 4.1: Representação do Sistema de Transporte de peças.....	49
Figura 4.2: Variáveis do Sistema de Transporte de Peças	51
Figura 4.3: Posições das Bases para Desligamento da Esteira 1	52
Figura 4.4: Diagrama de Relés do Controle do Motor da Esteira 1	53

Figura 4.5 : Posições das Bases em Relação a Trava 1	54
Figura 4.6: Diagrama de Relés do Controle da Trava 1.....	55
Figura 4.7: Layout para Operação do Elevador.....	55
Figura 4.8: Diagrama de Relés do Controle do Elevador	58
Figura 4.9: Grafcet do Controle do Motor da Esteira 1	59
Figura 4.10: Grafcet do Controle da Trava 1	60
Figura 4.11: Grafcet de Controle do Elevador	61
Figura 4.12: Grafcet do Processo de Montagem	63
Figura 4.13: Tela de Execução do Projeto Estação de Montagem/Desmontagem.....	66
Figura 4.14: Fotografia da Estação de Montagem/Desmontagem.....	67
Figura 4.15: Monitoração do Processo de Troca de Pallets	74
Figura 5.1: Estruturação do Sistema de Supervisão	77
Figura 5.2: Formulário para Emissão de Ordens de Serviço.....	78
Figura 5.3: Entradas/Saídas do Sistema de Supervisão	79
Figura 5.4: Etapas de Produção do Projeto PIPEFA	79
Figura 5.5: Conversão Base 10 para Base 3.....	82
Figura 5.6: Processos Simultâneos Realizados no PIPEFA.....	86
Figura 5.7: Estado das Entradas e Saídas do PIPEFA	89
Figura 5.8: Estado das Saídas do Projeto PIPEFA	89
Figura 5.9: Diagrama de Relés das Travas do PIPEFA.....	93
Figura 5.10: Diagrama de Relés dos Processos do PIPEFA	94
Figura 5.11: Sistema de Supervisão Desenvolvido a partir do P-CIM.....	97
Figura 5.12: Geração de Alarmes Utilizando-se P-CIM.....	98

Figura 5.13: Exemplo de Utilização do PL7 Micro.....	99
Figura 8.1: Grafcet do Sistema de Supervisão para o PIPEFA.....	110
Figura 8.2: Macros 1, 2 e 3 do Grafcet PIPEFA.....	111
Figura 8.3: Macros 4, 5 e 6 do Grafcet PIPEFA.....	112
Figura 8.4: Macros 7, 8 e 9 para o Grafcet PIPEFA.....	113
Figura 8.5: Especificação do conector Fêmea DB25.....	146
Figura 8.6: Link Serial com CLP Koyo.....	147
Figura 8.7: Protocolo de Comunicação Koyo.....	147
Figura 8.8: Aquisição de Dados da Porta Paralela.....	151
Figura 8.9: Aquisição de Dados do CLP da Koyo.....	152

Lista de Tabelas

Tabela 2.1: Características Físicas do Fieldbus.....	25
Tabela 4.1: Variáveis de Entrada e Saída para o Transporte de Peças	51
Tabela 4.2: Tabela de Estados para o Controle do Motor da Esteira 1.....	52
Tabela 4.3: Tabela de Estados para o Controle da Trava 1 (Stop 1).....	54
Tabela 4.4: Tabela de Estados para Operação do Elevador.....	57
Tabela 4.5: Variáveis de Entrada e Saída para a Estação de Montagem/Desmontagem..	64
Tabela 5.1: Banco de Dados Utilizado pelo Sistema de Supervisão	81
Tabela 5.2: Controle das Travas da Plataforma PIPEFA.....	91
Tabela 5.3: Operação dos Postos de Trabalho Controlados pelo CLP.....	92
Tabela 8.1: Especificação dos Códigos Decimal e Base três.....	107
Tabela 8.2: Relação de Entradas (CLP-PC).....	108
Tabela 8.3: Relação de Saídas (PC_CLP).....	109
Tabela 8.4: Especificação dos Pinos da Porta Paralela.....	144
Tabela 8.5: Portas Lógicas	158

Capítulo 1

Introdução

Este capítulo tem como objetivo justificar a importância da automação para as empresas, sua evolução histórica, seus problemas e as tendências para o futuro. Também são enfocados os objetivos e o delineamento do trabalho.

A arte de controlar é tão antiga quanto as necessidades humanas de desenvolver seus próprios sentidos. Mesmo não dispondo de grandes tecnologias, o homem mantinha a qualidade em seus artefatos e desenvolvia projetos eficazes direcionados ao controle de suas exigências. De acordo com o dicionário Aurélio (Ferreira, 1986), a palavra controle, de origem francesa (contrôle), denota o ato ou poder de domínio, fiscalização. De acordo com Silveira (1998), o início do controle na automação de processos surgiu logo após os primeiros reguladores mecânicos do tipo desenvolvido por James Watt em 1788, que utilizavam a instrumentação e reguladores do tipo pneumático e hidráulico e, após as guerras mundiais, do tipo eletrônico com tecnologia analógica. Ainda de acordo com Siveira (1998) o termo automatização se difundiu desde a construção das primeiras máquinas e se consolidou com a revolução industrial e, portanto, automatização está indissolúvelmente ligada à sugestão de movimento automático, repetitivo, mecânico e é sinônimo de mecanização, portanto reproduz ação. A automação é um conceito e um conjunto de técnicas por meio das quais se constroem sistemas ativos, capazes de atuar com uma eficiência ótima pelo uso de informações recebidas do meio sobre o qual atuam.

A produtividade, a confiabilidade e os custos são as principais características que as empresas estão visando para vencer a concorrência. Já houve época em que os produtos nacionais eram favorecidos por uma forte política de proteção, mas agora, com a globalização

da economia, as indústrias nacionais enfrentam uma forte concorrência das indústrias estrangeiras. Para vencer tal concorrência as indústrias necessitam melhorar a qualidade e baratear os custos de produção, não podendo mais adiar a automação de suas plantas de produção. Falar em automação, significa referir-se ao planejamento do processo, ao controle da qualidade do produtos, monitoração do funcionamento das máquinas, fluxo de materiais, supervisão do sistema e redes de comunicação.

A automação é uma área multidisciplinar, pois envolve três áreas de conhecimento: linguagem de programação (computação), plataforma eletrônica (eletrônica) e dispositivos de atuação (mecânica). Dessa forma, um estudo sobre automação é muito abrangente, envolvendo uma vasta gama de conhecimentos nas áreas de computação, eletrônica e mecânica.

A automação proporciona uma forte integração entre os setores de uma empresa, gerando um fluxo contínuo de informações entre todos seus departamentos. Esta integração corporativa esbarra num dos principais problemas da automação, que é a conectividade de seus equipamentos, principalmente os equipamentos ligados ao chão de fábrica, como os sensores, instrumentos inteligentes, controladores, máquinas CNC, etc.

Uma padronização neste sentido vem sendo buscada desde 1979, quando a ISO, “International Standards Organization”, iniciou os trabalhos para estabelecimento de um padrão para a interconexão de sistemas abertos (OSI – “Open Systems Interconnection”), o qual se tornou norma internacional em 1984.

A automação proporciona um controle de produção em tempo real (supervisão de processo). É possível obter informações a respeito de problemas e fluxo de produção em cada etapa do processo produtivo, informações estas, indispensáveis para aplicação de técnicas de controle de produção, como CIM (Computer Integrated Manufacturing) , CEP (Controle Estatístico de Processo), MRP (Manufacturing Resource Planning) e JIT (Just In Time).

O CIM estabelece o conceito da integração total da manufatura. Um ambiente CIM é um sistema complexo, constituído de redes de computadores, sistemas de banco de dados, protocolos para troca de dados e interfaces para aplicativos. O sistema CIM propõe a integração

de toda a empresa e da cadeia de fornecedores, desde o chão de fábrica até a gerência. De acordo com Silveira (1998), o CIM reflete muitos dos argumentos que uma empresa precisa para se tornar expoente no mercado em que atua, demonstrando ser uma empresa competitiva para sua época.

Neste trabalho será apresentado um estudo do sistema CIM, utilizando o projeto PIPEFA (Plataforma Industrial para Pesquisa, Ensino e Formação em Automação) como plataforma de desenvolvimento, descrito por Rosário (1996). Participam do projeto PIPEFA o LAR (Laboratório de Automação Integrada e Robótica) da UNICAMP, o IA (Instituto de Automação) do CTI (Centro Tecnológico para Informática) e o LIISI (Laboratoire d'Ingénierie Intégrée des Systèmes Industriels) do ISMCM-CESTI da França. O projeto PIPEFA visa o desenvolvimento de um ambiente que contemple vários aspectos de um sistema integrado de produção para a aplicação e teste de metodologias, tecnologias e componentes para sistemas CIM.

Neste estudo de automação, serão abordados os aspectos de programação (software) ligados à automação. Será vista a parte de programação dos atuadores e desenvolvimento de software para supervisão de processos. Será analisada a automação de processos discretos e a automação da manufatura.

Este estudo será feito com base na proposição de sistemas abertos, ou seja, adaptáveis a qualquer processo e utilizando-se protocolos de comunicação padrão. Neste trabalho são descritos ou utilizados os padrões RS232C Ethernet e interface paralela padrão Centronics™ (Tompson, 1991). Também são abordados aspectos de padrões que poderão ser utilizados em futuros trabalhos, como o USB (Universal Serial Bus) e Fieldbus (Barramento de Campo).

Nesse sentido serão buscadas alternativas para suporte de um conjunto consistente de padrões, referentes à tecnologia da informação, de forma a garantir a conectividade de todos os equipamentos envolvidos na automação.

1.1 Objetivos

O objetivo do trabalho é o estabelecimento de conceitos, metodologias, estratégias e ferramentas para a automação de processos discretos. Para tanto deverão ser cumpridas as seguintes etapas:

1. Inicialmente será feita a montagem de uma estrutura de aprendizagem e validação de programas para Sistemas Automatizados de Produção (SAP);
2. O passo seguinte será a escolha e testes de uma linguagem de programação, de baixo custo e amplamente difundida pelo mercado, para desenvolvimento dos programas didáticos de automação;
3. O último passo será o desenvolvimento de um algoritmo de supervisão capaz de promover uma integração que automatize as aplicações de processos, gerando e gerenciando tabelas em base de dados, construindo gráficos e relatórios de modo a promover um padrão de integração industrial em tempo real. Este algoritmo de supervisão deverá integrar o planejamento de recursos da empresa e o chão de fábrica através da troca de dados e informações entre aplicativos.

1.2 Delineamento do Trabalho

O trabalho está dividido na seguinte forma:

- Capítulo 1, Introdução, é feita uma consideração histórica sobre automação e uma apresentação geral do trabalho, incluindo os objetivos e delineamento do trabalho;
- Capítulo 2, Revisão Bibliográfica, são descritos os principais conceitos, propiciando o posicionamento diante do problema apresentado;

- Capítulo 3, Descrição do Projeto PIPEFA, são descritos detalhadamente todos os elementos constituintes do projeto da Plataforma Industrial para Pesquisa, Ensino e Formação em Automação (PIPEFA);
- Capítulo 4, Utilização de Sistemas de Supervisão em Automação, são analisadas e ensaiadas ferramentas para modelagem de sistemas seqüenciais, como Diagrama de Relés e Grafset;
- Capítulo 5, Implementação do Sistema de Supervisão ao Projeto PIPEFA, são utilizados os resultados da parte anterior para implementação de um sistema supervisor para controle do projeto PIPEFA;
- Capítulo 6, Conclusões e Perspectivas Futuras, são apresentadas as considerações finais e sugestões para os trabalhos posteriores.

Capítulo 2

Revisão Bibliográfica

Neste capítulo é dado suporte teórico para justificar a metodologia adotada no desenvolvimento das aplicações a sistemas automatizados de produção. É apresentado o conceito de Integração Total da Manufatura e as considerações acerca de seus problemas, assim como suas possíveis soluções. É apresentada a importância dos métodos para descrição de sistemas sequenciais, e sua relação com a estruturação da parte de comando de sistemas automatizados de produção.

Ao final deste capítulo será discutido aspectos ligados a área de software, como linguagem de programação e hardware; será abordado o desenvolvimento de sistemas de supervisão, que é a parte mais estreita da relação entre sistemas automatizados de produção com a área de software.

2.1 Conceitos Básicos

Sabe-se que o processo produtivo vem passando por profundas mudanças nos últimos anos. A extrema divisão do trabalho proposta pelo modelo Taylorista já não se mostra tão eficiente. Scheer (1994) alerta que apesar dos grandes benefícios na aceleração do manuseio de produtos, trazidos com a extrema especialização do trabalho (na qual o Taylorismo se baseia), a excessiva subdivisão do processo produtivo, acaba causando um aumento no tempo de transferência de dados e produtos. Mais especificamente, tem-se uma demanda maior em processos de comunicação (com ordens administrativas e de produção) e maiores tempos mortos no processo produtivo.

De acordo com Scheer (1994), a utilização de um sistema integrado pode melhorar o processo de transferência de informações. A utilização do sistema de Manufatura Integrada por Computador – CIM (“Computer Integrated Manufacturing”) permite um melhor fluxo de informações na cadeia produtiva. O CIM objetiva a integração, por meio da tecnologia da informação, de atividades realizadas por homens e máquinas através da comunicação, cooperação e coordenação de funções técnicas, administrativas e de suporte, utilizando redes de computadores, sistemas de banco de dados, formatos para troca de dados e interfaces para aplicativos.

De acordo com Teicholz (1989) o sistema CIM é necessário para eliminar a grande quantidade de papéis dentro das indústrias, e também para eliminar trabalhos desnecessários. Além disto, o CIM integra os sistemas CAE/CAD/CAM, impedindo que se transformem em “ilhas de automação”, onde processos individuais são automatizados sem a preocupação de integração com os demais processos. Teicholz (1989) sugere uma concepção genérica do sistema CIM como mostrado na Figura 2.1, mas alerta que esta concepção deve ser adaptada à cada empresa, de forma a obter-se a solução mais eficiente, a qual reflita um balanço individual entre cada elemento da empresa.

Para melhor compreender a estrutura do sistema CIM proposto por Scheer (1994), é possível subdividi-lo nos seguintes componentes:

- PPC, Production Planning and Control, componente do CIM responsável pelo planejamento e controle da produção;
- CAD, Computer Aided Design, componente do CIM responsável pela modelagem geométrica do produto;
- CAE, Computer Aided Engineering, componente do CIM que trata das especificações tecnológicas para a definição do produto;
- CAPP, Computer Aided Process Planning, componente do CIM responsável pela otimização do planejamento de processos;

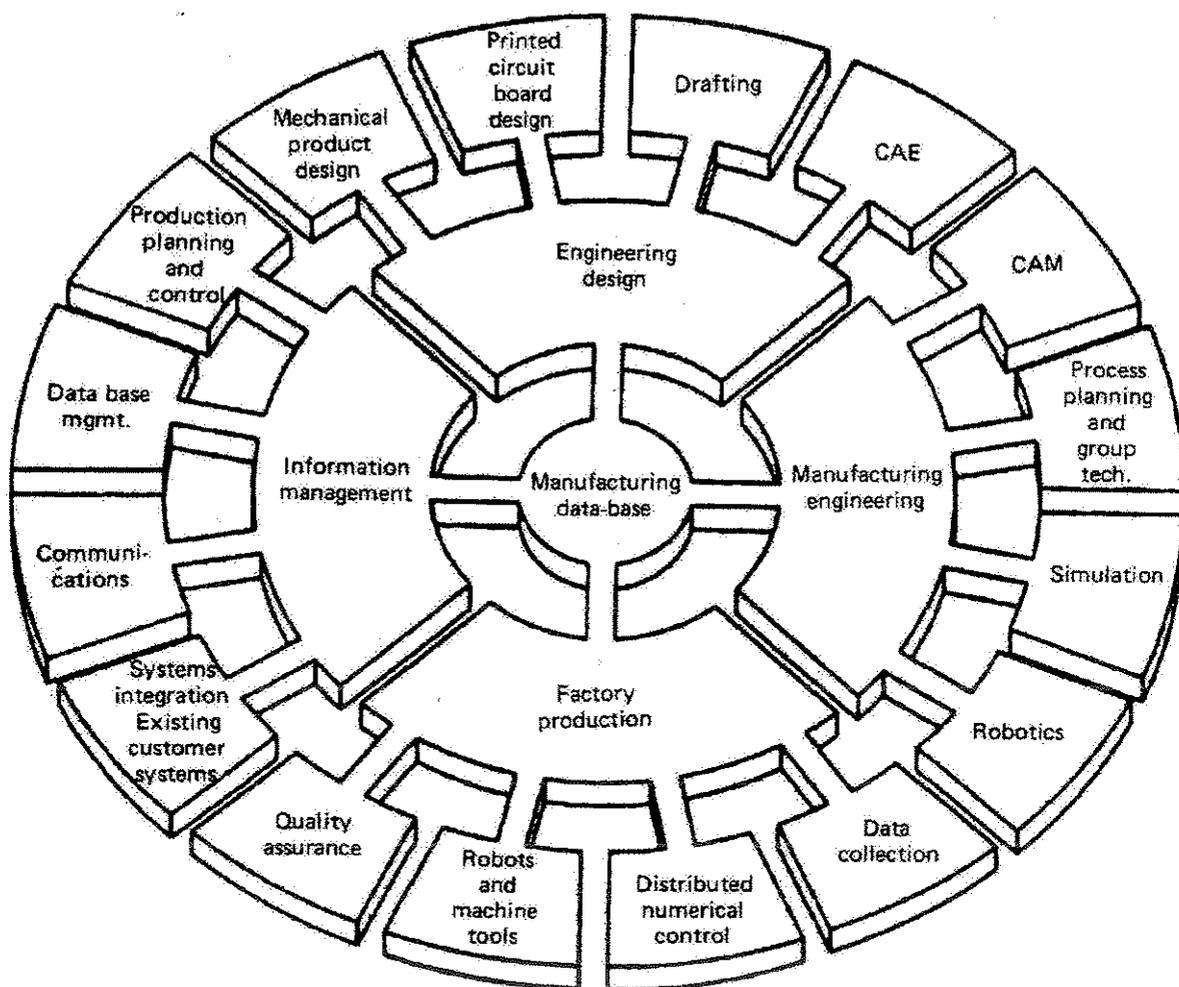


Figura 2.1: Arquitetura Genérica para Sistemas CIM (Teicholz, 1989)

- CAP, Computer Aided Planning, componente do CIM responsável pela Engenharia de Desenvolvimento de Produção;
- CAM, Computer Aided Manufacturing, componente do CIM responsável por transformar o projeto do produto em um programa de manufatura;
- CAQ, Computer Aided Quality Assurance, componente do CIM responsável pela Garantia da Qualidade;

- Manutenção, componente do CIM responsável por assegurar o funcionamento das máquinas e equipamentos.

Esta forma de divisão do CIM, visa aproximar de forma direta a esfera administrativa e a área produtiva das organizações, munindo-se com flexibilidade e com instrumentos de controle muito eficazes, através da agilização do fluxo interno de dados e de um manejo condizente. De maneira simplificada, pode-se segmentar o sistema CIM, de acordo com Scheer (1994), em três níveis hierárquicos, como mostrado na Figura 2.2:

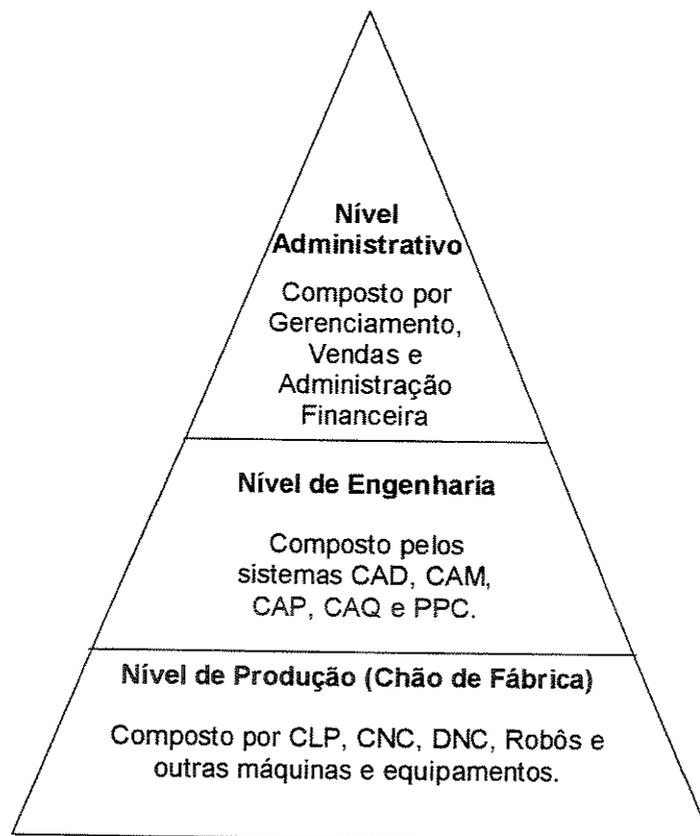


Figura 2.2: Hierarquia do Sistema CIM

- nível Administrativo Geral;
- nível de Engenharia;
- nível de Produção;

Outros autores também estratificam o CIM em três níveis, mas Silveira (1998) dá uma conotação um pouco diferente para cada nível. Para ele os três níveis que compõem o CIM são:

1. nível de planejamento, responsável pela coordenação, gerenciamento e controle de toda a empresa, baseado em análises e viabilidades que tomam decisões a médio e a longo prazos. Define o desenvolvimento de novos produtos, gera pedidos de fabricação e estabelece uma atuação estratégica junto ao departamento de marketing;
2. nível de controle, no qual são tomadas decisões de curto prazo, relacionadas às decisões de tarefas e eventos, além de envio, execuções e acompanhamento dos pedidos de fabricação;
3. nível de processo, a partir das decisões do nível anterior, realiza a fabricação propriamente dita, com deliberação em tempo real.

Todo conjunto de tarefas executadas por um computador, a fim de obter um sistema formal que gerencie o fluxo de materiais na fase de processo, pode ser classificado como técnica MRP (Manufacturing Resource Planning). Tal sistema está disposto a informar a necessidade de componentes para a produção de produtos acabados, bem como de suas datas de necessidade. De acordo com Silveira (1998), a integração do MRP com os outros níveis é realizada pelo MES (Manufacturing Execution System). Um empreendimento dotado do sistema MES possui maior controle, dentro dos níveis hierárquicos, de todas as informações geradas de forma a tornar mais eficiente toda sua comunicação. Como o sistema atua com o objetivo de integração de todos os níveis hierárquicos, todos os programas baseados nesse sistema possuem conectividade com um banco de dados relacional, empregando uma estrutura típica de cliente/servidor.

Cabe ressaltar que cada nível do modelo CIM possui equipamentos de diferentes naturezas, por exemplo: no nível superior estão os microcomputadores, no nível intermediário, as estações de trabalho CAE/CAD/CAM e no nível inferior, os equipamentos programáveis do chão de fábrica.

Como pode ser observado, os equipamentos que compõem o CIM são bastante heterogêneos, dificultando a conectividade entre os níveis e, mais especificamente, no nível

inferior, onde os equipamentos de instrumentação, os CLPs, os CNCs, os SDCDs (Sistemas Digitais de Controle Distribuído) e os Controles de Robôs não possuem ainda um protocolo padrão de comunicação.

Dentro dos aspectos do CIM tratados por Scheer (1994), o de maior importância para este trabalho é o conceito de MAP, “Manufacturing Automation Protocol”. Ele trata da dificuldade de integração dos diversos componentes do CIM, devido à ausência de um protocolo de comunicação padronizado e eficiente.

O protocolo é definido como um conjunto de regras para o estabelecimento de comunicação entre sistemas. De acordo com Tanenbaum (1994), uma padronização para a criação de protocolos foi proposto pela ISO, “Internacional Standards Organization”, em 1979, com o padrão para a interconexão de sistemas abertos (OSI – “Open Systems Interconnection”). O modelo de referência OSI está estruturado em sete camadas distintas, bem definidas e independentes, descritas por Aguiar (1997) como:

1. Física, destinada a certificar a transmissão e recepção corretas dos bits, definindo vários fatores como: tensão do pulso, duração, pinos do conector de rede, entre outros.
2. Enlace de Dados, especificada para correção de erros de transmissão de dados.
3. Rede, destinada ao encaminhamento, controle de fluxo de pacotes e endereçamento.
4. Transporte, que tem a função de prestação de serviços de conexão, como por exemplo o estabelecimento da conexão, transferência de dados e finalização da sessão.
5. Sessão, otimiza a troca de dados entre entidades, e gerencia o diálogo das duas máquinas.
6. Apresentação, faz a interpretação e manutenção da sintaxe e semântica quando da execução de aplicações remotas.

7. Aplicação, faz a conversão entre os diversos tipos de terminais, controles de operações, mapeamentos de memória e controle de transferência de arquivos.

O modelo de referência OSI será utilizado para o desenvolvimento de aplicativos que trabalham com transferência de dados entre os componentes do CIM.

O nível inferior do CIM, nível de produção, é composto pelas máquinas e equipamentos responsáveis pela produção. Estes equipamentos automatizados necessitam ser programados de acordo com o estabelecido pelo nível de Engenharia. A seguir será visto como deve ser realizada a programação destes equipamentos.

2.2 Descrição de Sistemas Seqüenciais

Para descrever o funcionamento de um sistema automatizado, utiliza-se o caderno de especificações. Todo o desenvolvimento do sistema automatizado é baseado no caderno de especificações, pois nele constam todas as etapas previstas para o processo de automação, logo, o primeiro passo para automatizar um sistema é a elaboração do caderno de especificações. De acordo com Bittar (1993), a elaboração do caderno de especificações deve ser feita em dois níveis complementares:

- primeiro nível ou especificações funcionais, onde é descrito o comportamento da parte de comando frente a parte operativa, ou seja, as funções que o automatismo deva desempenhar perante as diversas situações que possam ocorrer;
- segundo nível ou especificações tecnológicas, quando são acrescentadas especificações tecnológicas e materiais às especificações funcionais, ou seja, são acrescentadas informações ao nível anterior, como: natureza dos sensores e atuadores, faixas de temperatura, umidade, pressão, nível de poeira, ou outras especificações que possam afetar o processo.

Para passar as informações do caderno de especificações para a parte de comando, é utilizado um instrumento de representação, pois a linguagem corrente não se mostra apta a tal tarefa. Tal instrumento de representação irá transcrever as informações contidas no caderno de especificação, para uma linguagem que possa ser entendida pela parte de comando.

Para melhor compreender como tornar o sistema operacional, ele será dividida em duas partes, a saber:

- parte operativa;
- parte de comando.

A parte operativa é a parte física do sistema, ou seja, é a responsável pela execução das operações do sistema. Ela é composta pelos atuadores, sensores e outras partes físicas do sistema, como molas, punções, ferramentas de corte, bombas, soldadores, marcadores, etc.

A parte de comando é a parte responsável pelo controle do sistema. Ela troca informações com o sistema e com o meio exterior, gerando sinais de aviso e controlando a parte de comando. Por exemplo, num elevador, os botões de controle, o indicador eletrônico de andar e o sinal de emergência representam a parte de comando, enquanto os motores, a cabina e os cabos de aço representam a parte operativa, como mostrado na Figura 2.3:

Graças a troca de informações com a parte operativa, é que a parte de comando é mantida informada do estado do avanço das operações. Os sinais de todos os sensores e para todos os atuadores devem ser conectados a parte de comando. Levando-se em conta que, todos os sensores e atuadores devem estar conectados à parte de comando, pode-se ter idéia da quantidade de cabos e ligações a serem realizadas. Isto demonstra a grande importância da comunicação de dados em sistemas automatizados.

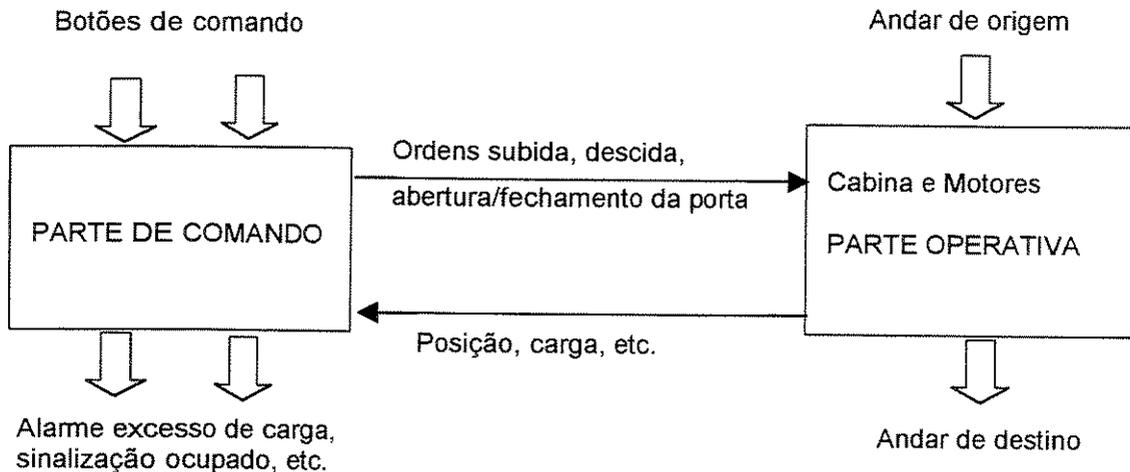


Figura 2.3: Parte de Comando e Operativa de um Elevador

2.3 Controlador Lógico Programável

As primeiras técnicas que surgiram para automatizar máquinas e equipamentos utilizavam lógicas muito extensas e complicadas, e para executá-las eram usados relés, contadores e temporizadores eletromecânicos. Esses equipamentos além de caros, tinham uma vida útil curta. Eles utilizavam uma linguagem de programação chamada de Diagrama de Relés (“Ladder Diagram”) pois tinham o sistema de Quadros de Relés como *hardware* básico.

Somente com a introdução do microprocessador no controle de processos pode-se simplificar este processo de programação, evitando-se as trabalhosas substituições nos quadros de comando. A este equipamento microprocessado deu-se o nome de Controlador Lógico Programável.

O Controlador Lógico Programável (CLP)¹ ou Controlador Programável (CP) é definido pelo Industrial Control and Systems (1998), como um equipamento de eletrônica digital, que utiliza uma memória programável para armazenamento interno de instruções, as quais serão

¹ PLC (“Programmable Logic Controller”) é marca registrada de Allen-Bradley Company

utilizadas para implementação de determinadas funções, como: lógica, seqüenciamento, temporização, contagem e aritmética para controle. Para isto, são utilizados módulos de entrada e saída, aplicáveis a vários tipos de máquinas e processos.

De acordo com Murata (1995) existem três tipos de modelos de especificação de seqüência de controle. O primeiro e mais simples é o Diagrama de Estado-Transição. O segundo tipo é o modelo baseado em regras, onde se enquadra o Diagrama de Relés. No terceiro tipo estão os modelos baseados em Redes de Petri, onde se enquadram a maioria das linguagens de programação de CLP utilizadas atualmente. A partir destes três tipos de modelos, muitas entidades e fabricantes têm desenvolvido suas próprias linguagens, como o Step 5, Grafcet e CSF. A maioria dessas linguagens são variações do “Sequential Function Chart” (SFC), uma linguagem de programação de CLPs normalizada pela International Electrotechnical Commission (1993), IEC 1131-3. Essas linguagens derivadas do SFC, programam a parte de comando de uma maneira contínua, na qual através de uma representação gráfica na forma de Diagrama de Blocos, tem-se uma descrição quase natural do caderno de especificações.

De acordo com Brendel (1992), o SFC possibilita a organização da programação em uma série de etapas e transições intercaladas por ligações orientadas. As etapas representam a situação na qual as entradas e saídas executam as ações. As etapas são representadas graficamente por blocos. As ligações entre as etapas são representadas graficamente por linhas verticais de união. As transições representam a condição onde o controle passa de uma etapa a outra. São representadas graficamente por uma pequena linha horizontal cruzando a linha da ligação, como mostrado na Figura 2.4. Cada transição deve ser associada com uma expressão Booleana, que só dará prosseguimento a próxima etapa do programa caso a expressão seja verdadeira.

A representação do SFC em Grafcet é definida por Murata (1995) como uma subclasse Condição/Evento do modelo de Redes de Petri. Mas, segundo Murata (1995), alguns autores afirmam que o Grafcet é uma variação do Diagrama de Estado-Transição, na qual foi acrescentada a opção de paralelismo. De qualquer forma, a representação do SFC em Grafcet é semelhante com a representação em Redes de Petri (DiCesare,1993), com a diferença básica de

possuir maior número de mensagens relacionadas às entradas e ao ambiente. O Grafcet é mais utilizado nas indústrias do que as Redes de Petri, apesar de ser mais difícil sua validação, devido a maior quantidade de informações.

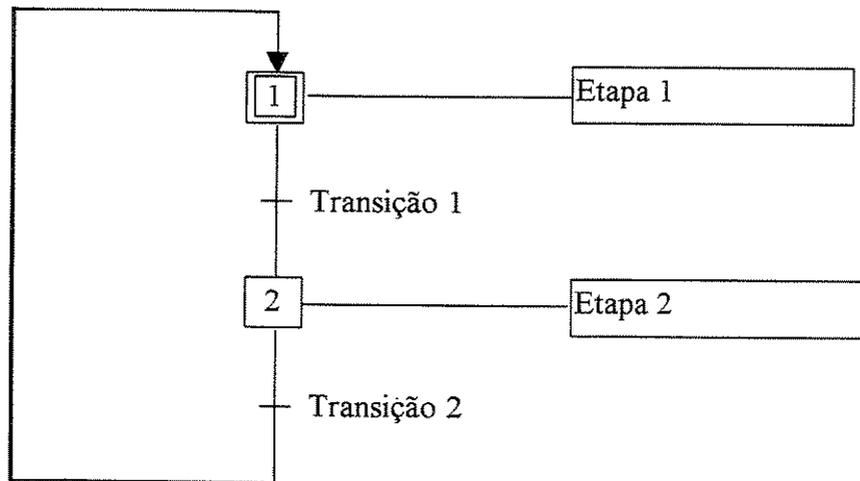


Figura 2.4: Exemplo de SFC

De acordo com a TELEMEC (1986), a denominação Grafcet “Grafo de Comando Etapa-Transição”, foi apresentada em 1977 pela AFCET (Associação Francesa para a Cibernética Econômica e Técnica), fruto de estudos sobre o modo de unificar a representação do caderno de encargos de um automatismo lógico, iniciados em 1975. Este método é um Diagrama Funcional, isto é, uma representação gráfica, logo concisa e de fácil leitura, que permite descrever as funções realizadas pelo automatismo. Ainda de acordo com a TELEMEC (1986), para distingui-lo dos outros diagramas funcionais muito correntemente utilizados na indústria, mas não beneficiando de um tal suporte teórico nem de um tal consenso, era necessário dar-lhe um nome próprio. O nome Grafcet foi escolhido, lembrando assim a sua origem e colocando bem em evidência a sua originalidade.

2.4 Diagrama de Relés versus Diagrama Seqüencial

Conforme relata David (1995), a representação do caderno de especificações através de Diagrama de Relés é uma tarefa bastante árdua. Primeiro deve-se construir uma Tabela de Estados, onde são representados os estados de todos os sensores e atuadores durante o processo.

A partir da Tabela de Estados escreve-se as lógicas da programação, onde cada linha da Tabela de Estados irá corresponder a uma lógica. A Tabela de Estados não pode ter linhas idênticas, e se isto ocorrer, deve-se criar variáveis adicionais que modifiquem estas linhas. Dessa forma, o programa não obedece a um fluxograma e, uma vez satisfeitas as condições de entrada, a parte de comando executará a condição de saída, independentemente da seqüência das linhas de programação.

Já a programação através do Diagrama Seqüencial obedece a um fluxograma, de forma que a próxima etapa do programa só será executada após a execução da etapa anterior, como relata Rosário (1997). O programa feito desta forma fica semelhante ao caderno de especificações, onde as etapas de programação são feitas de uma forma seqüencial.

2.5 Automação da Manufatura

O processo de automação da manufatura pode ser classificado de duas maneiras:

- Sistema de Automação rígida;
- Sistema de Automação flexível.

O sistema de automação rígido é aquele onde as funções predefinidas, que o equipamento deverá executar são muito difíceis e até impossíveis de serem alteradas.

O sistema de automação flexível é caracterizado pela facilidade na alteração das funções predefinidas. Com isso, uma linha de montagem pode ser rapidamente preparada para produzir outro produto. Um sistema de automação flexível é normalmente composto por robôs e máquinas CNC (Comando Numérico Computadorizado).

Os novos sistemas computadorizados de manufatura vem se sobrepondo, de forma definitiva, aos antigos sistemas de automação rígida. As novas necessidades de produção rápida

de lotes pequenos e diversificados, impostas pela filosofia “just in time”, demonstra a grande importância das células flexíveis automatizadas de manufatura.

As primeiras máquinas de Comando Numérico (CN) que surgiram eram capazes apenas de reproduzir seqüências de usinagens gravadas em fitas perfuradas. Só quando o computador foi interligado com o CN, é que passou a utilizar-se os recursos da simulação gráfica, edição de programas na própria máquina e gravação de dados em disquetes. Este tipo de comando passou a se chamar Comando Numérico Computadorizado (CNC).

As máquinas CNC são empregadas na produção de pequenos e médios lotes de peças, quando se necessita uma rápida adaptação do sistema durante a usinagem de lotes diversificados.

Devido a grande variedade de fabricantes de Comandos Numéricos, há uma enorme dificuldade de comunicação entre as máquinas e dispositivos externos. Até mesmo na linguagem de programação, que apesar dos fabricantes afirmarem que estão trabalhando de acordo com a norma ISO 6983 e DIN 66025, um programa feito para a máquina de um fabricante não funciona na máquina de outro fabricante, não garantindo o intercambiamento de programas entre máquinas de diferentes fabricantes.

Uma característica marcante nos novos equipamentos de informática é a tecnologia “plug and play”. A Intel (1998) promete com a nova porta de comunicação para PC e Apple, a USB (Universal Serial Bus), interligar até 127 periféricos em um único barramento. Esta interligação será feita com tecnologia “plug and play”, sem necessidade de abrir o computador para adição de cartões, sem “DIP switch” para serem configurados e sem a necessidade de reservar Requisições de Interrupção (IRQ). A ligação dos periféricos é feita em forma de cascata, ligando-se um periférico após outro no mesmo cabo. O barramento USB padrão possui taxa de transferência de 12Mb/s, para uma distância de até 5m.

Esta tendência também deverá se refletir nos novos equipamentos CNC. No futuro, para interligar um robô a uma máquina CNC, será necessário apenas fazer a ligação dos cabos de força e comunicação.

Para que a Unidade Central de Processamento (CPU) da máquina CNC, desempenhe sua função com rapidez e segurança, é necessário a existência de uma interface de entrada/saída (Interface I/O) entre ela e os componentes por ela comandados. Esta interface, além de fazer a proteção do sistema, permite trabalhos com diferentes níveis lógicos. Esta interface é programada em linguagem de Relés, interfaceando os sinais de entrada ou saída para torná-los compatíveis com os níveis e frequência necessária para utilização pela CPU. Por estas características esta interface é classificada como Controlador Programável da máquina CNC.

Uma máquina CNC possui basicamente dois tipos de controles. O primeiro tipo é o controle de posicionamento dos eixos, e o segundo é o controle dos dispositivos periféricos, ou seja, é a interação do movimento dos eixos com os outros dispositivos da máquina. Dessa forma, o processamento das informações necessárias ao funcionamento da máquina é feito de forma paralela, onde tem-se um controle para o movimento dos eixos, e outro para a interação dos movimentos dos eixos com outros dispositivos da máquina, realizado pelo CLP (FANUC 1990,1991 e GE 1988).

De acordo com a ROMI (1993 e 1995), os dispositivos controlados pelo CLP são:

- Sensores de fim de curso;
- Controle de fluido refrigerante;
- Controle de óleo lubrificante;
- Transportador de cavacos;
- Troca de ferramentas;
- Movimentação do magazine de ferramentas;
- Troca de pallets;

- Controle da rotação do eixo árvore;
- Habilitação ou interrupção da execução do programa;
- Abertura ou fechamento das portas (principal, pré-seleção e pallet);
- Gerenciamento da carga sobre o sistema;
- Controle do sistema hidráulico;
- Controle do divisor para mesa indexada;
- Controle do sistema de pré-ajuste das ferramentas;
- Sinal de emergência e alarme sonoro;
- Controle de limpeza de ferramenta;
- Controle dos estados da máquina:
 - Trabalho;
 - Atenção;
 - Parada;
 - Permite partida.
- Controle do freio do eixo Z e divisor.

Através dos sinais de Entrada/Saída do CLP pode-se, de acordo com ROMI (1993), acoplar robôs ou outros dispositivos de carga e descarga ao CNC. Pode-se também utilizar estes

sinais para desenvolver um sistema de supervisão capaz de controlar e monitorar uma máquina CNC a distância.

2.6 Sistema de Supervisão

De acordo com o dicionário Aurélio (Ferreira, 1986), supervisão é ação ou efeito de supervisionar ou supervisionar, que por sua vez significa dirigir, orientar ou inspecionar em plano superior. Ainda de acordo com o Aurélio, monitorar ou monitorizar significa acompanhar e avaliar (dados fornecidos por aparelhagem técnica), restritivamente pode significar controlar, mediante monitorização. De qualquer forma o termo monitorar está mais ligado a observação e registro, enquanto supervisionar está mais ligado à direção, administração, condução.

Muitas vezes o termo Sistema de Supervisão é confundido com Controle de Processos. Como relata Jafari (1995), é possível, mas não eficiente, desenvolver estas duas atividades na mesma unidade. O Controle de Processos, por definição, é o responsável pelo funcionamento do processo, o qual normalmente é contínuo. A função do Sistema de Supervisão, contudo, está ligada com a transição de estado dos processos individuais, como a coordenação entre esses processos. Pode-se dizer que o Sistema de Supervisão tem uma visão discreta do sistema, enquanto o controle de Processo tem uma visão contínua do sistema. Ainda segundo Jafari (1995), o Sistema de Supervisão pode ser em malha aberta ou fechada. No caso de malha fechada, o Sistema de Supervisão controla ações, influenciado pelos sensores de entrada, que monitoram a planta de processo.

Para a supervisão de uma planta automatizada, de acordo com Bolton (1995), pode-se utilizar três formas de controle de processos, o Sistema de Controle Centralizado, o Hierárquico e o Distribuído.

No Sistema de Controle Centralizado todo o controle do processo é feito por um único equipamento. Este equipamento tem acesso direto a todas as variáveis do processo. Ele processa um algoritmo pesado, que faz o controle do processo, verificações de intertravamento, cálculos,

armazenamento de dados e geração de alarmes. Se este equipamento falha, todo o sistema falhará. Este tipo de controle foi muito usado na década de 60.

No sistema de controle Hierárquico, existe uma ordenação de autoridade entre os componentes do sistema, de acordo com as funções que eles realizam. Por exemplo, os computadores que manipulam um grande número de tarefas, são supervisionados pelos computadores de uma camada de nível superior.

No sistema de controle distribuído, diversos componentes do sistema executam tarefas similares. Dessa forma, quando há uma falha em um componente do sistema, sua tarefa é automaticamente transferida para outro componente.

De acordo com Bolton (1995), o tipo de controle mais utilizado atualmente é uma mistura do controle Hierárquico com o Distribuído. Neste tipo de sistema, tem-se uma ou várias camadas que supervisionam o funcionamento das camadas inferiores. Se algum componente falha, a camada superior transfere suas funções para outros componentes. Por exemplo, no nosso caso ter-se-á a camada inferior sendo composta pelos CLPs escravos, uma camada intermediária composta pelo CLP mestre e uma camada superior composta pelo computador com o sistema de supervisão. Assim, se um CLP escravo falhar, suas funções poderão ser transferidas para outro componente do sistema.

Dessa forma, o sistema de supervisão apenas verifica o funcionamento de cada sub-processo, deixando a parte do controle e geração de alarmes por conta dos CLPs. Neste caso, o Sistema de Supervisão é responsável apenas pela verificação, cálculos e armazenamento de dados. A verificação de intertravamento entre os diversos sub-processos é feita por um CLP Mestre, que é o elo de ligação entre os CLPs escravos e o Sistema de Supervisão, como mostrado na Figura 2.5. As vantagens deste Sistema de Supervisão são:

- Maior rapidez do sistema, uma vez que tem-se o controle repartido por vários equipamentos;

- Maior segurança, pois o dano em um dos equipamentos do sistema não afeta os demais;
- Maior versatilidade, uma vez que tem-se vários pequenos algoritmos é mais fácil o desenvolvimento e implementação de mudanças;
- Menor tráfego de informações, uma vez que tem-se o fluxo só da informações importantes;
- Facilidade de manutenção;
- Não sobrecarrega o sistema.

A comunicação entre os CLPs pode ser feita utilizando-se protocolos de comunicação proprietários, ou protocolos abertos, como mostrado na Figura 2.5. Uma padronização para comunicação entre CLPs está sendo proposta pela Fieldbus Foundation através de Barramento de Campo (Fieldbus), e deverá ser implementada em breve. De acordo com Souza (1996), a Fieldbus Foundation está viabilizando uma solução comercial utilizando como base as propostas do IEC/SP-50, PROFIBUS PA, WorldFIP e HART. De acordo com Peluso (1997), a utilização do padrão RS232C para comunicação entre os diversos componentes do sistema utilizará uma grande quantidade de fios, gerando um excessivo número de ligações de cabos. A utilização de um protocolo proprietário gerará uma dependência de hardware, pois deverão ser adquiridos equipamentos de um mesmo fabricante. Dessa forma, a melhor solução é a utilização do protocolo Fieldbus.

O protocolo Fieldbus é um sistema de comunicação digital bidirecional, que interliga equipamentos inteligentes de forma serial através de um único meio físico. Fieldbus é um protocolo interoperável baseado no modelo OSI, suportado pela quase totalidade dos fabricantes mundiais de instrumentação. De acordo com Peluso (1997), a opção de baixa velocidade para Fieldbus é 25 vezes mais rápida que os protocolos comuns para transmissores inteligentes, além de ser muito mais eficiente. Ele permite a comunicação entre uma variedade de equipamentos, como: transmissores, válvulas, controladores, CLPs, leitora de código de barras, indicadores

dedicados, programadores e registradores. Fieldbus é uma tecnologia nova, que ainda não está padronizada. Ele ainda apresenta algumas divergências conceituais, dependendo da corrente tecnológica utilizada, como Fieldbus Foundation, PROFIBUS, WorldFIP e IEC-ISA/SP50.

De acordo com Souza (1996), o modelo OSI, no qual o Fieldbus se baseia, consiste de sete níveis, como mencionado anteriormente. Porém, para aplicação em tempo real, os níveis de 3 a 6 não são considerados, já que eles tratam da transferência de dados entre redes. Para a aplicação do Fieldbus, são usados os seguintes níveis:

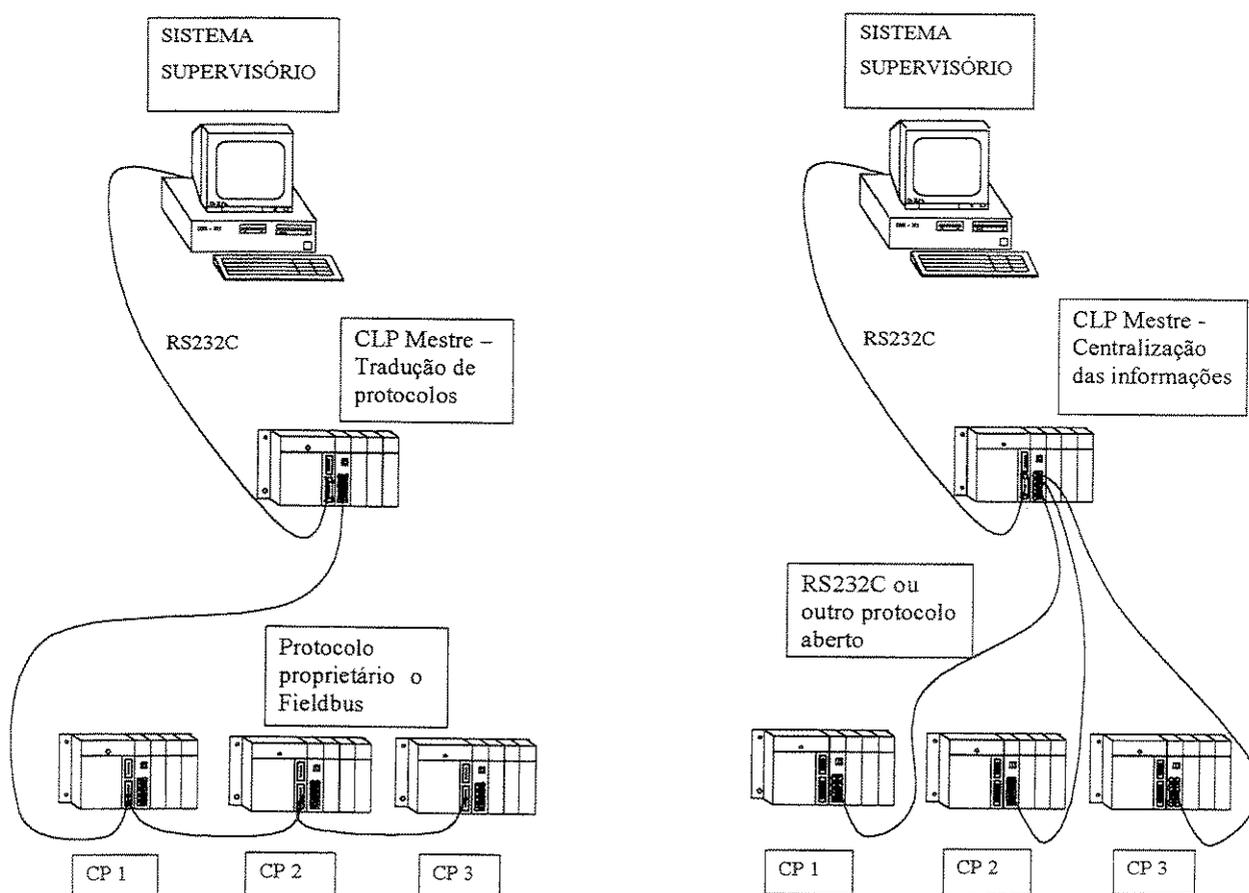


Figura 2.5: Layout para Rede de Comunicação de CLPs

- Nível 1 – Nível Físico. Define o tipo de sinal de transmissão, o meio físico para transmissão, velocidade da transmissão de dados, etc.
- Nível 2 – Nível de Enlace. Define a interface entre o nível físico e o nível aplicativo. Ele estabelece como as mensagens devem ser estruturadas, garante a integridade das mensagens e controla o acesso à rede.
- Nível 7 – Nível de Aplicação. Define como os dados e comandos são especificados, endereçamento e representação.

Como o Fieldbus ainda não está padronizado, dependendo da tecnologia de desenvolvimento ele pode aceitar diferentes topologias de rede, meios físicos, número de equipamentos e distância máxima de transmissão, como relata a Synergetic (1997), e mostrado na Tabela 2.1.

Tabela 2.1: Características Físicas do Fieldbus

Tecnologia	Topologia da rede	Meio físico	Max. Equip. (nós)	Max. Dist
Profibus	linha, estrela e anel	par trançado ou fibra ótica	127 nós	24 km (fibra)
Fieldbus Foundation	multiponto c/ barramento	par trançado	240 por segmento	1900m - 31.25k 500m - 2.5Mbps
IEC/ISA SP50	estrela ou barramento	par trançado, fibra e rádio	128 nós	1700m - 31.25k 500m - 5Mbps
WorldFIP	barramento	par trançado ou fibra ótica	256 nós	Até 40 km

2.7 Ambiente de Programação

Para desenvolvimento de aplicativos pode-se utilizar plataformas Risc, ou microcomputadores no padrão Apple ou IBM-PC. A plataforma Risc é mais indicada para sistemas que necessitam de altos desempenhos, pois tem alto poder de processamento, mas com um custo elevado. O microcomputador padrão Apple é mais indicado para trabalhos multimídia. O microcomputador padrão IBM-PC é o mais indicado para sistemas de baixo custo, como nosso caso, pois além de apresentar menor custo, é amplamente utilizado no mercado mundial, possuindo muitas interfaces, periféricos e opções de software. Para Sistema Operacional de um equipamento IBM-PC, pode-se optar por OS/2®, Windows95® ou WindowsNT®. Por serem menos susceptíveis a falhas, o WindowsNT® e o OS/2® poderiam até ser uma boa opção, no entanto optou-se pelo Microsoft Windows95®, por ser o mais difundido. É importante trabalhar com um Sistema Operacional utilizado pela grande maioria dos usuários, pois isto facilita o desenvolvimento de aplicativos e a continuidade deste trabalho.

A linguagem de programação deve permitir o desenvolvimento rápido de aplicativos Windows® (RAD - “Rapid Application Development”), com facilidade de utilização da interface gráfica de usuário (GUI – Graphic User Interface). Neste tipo de linguagem de programação, segundo o Guia do Programador (1997), ao invés de escrever muitas linhas de código de programação, para descrever a aparência e localização de elementos da interface, simplesmente se arrasta e solta objetos pré-montados em locais da tela.

As duas linguagens RAD mais difundidas são Borland Delphi e o Microsoft Visual Basic. Elas trazem diferenças estruturais derivadas das linguagens que as deram origem. Enquanto o Delphi evoluiu do Borland Pascal para DOS, o Visual Basic evoluiu do Microsoft Basic (Beginners All-purpose Symbolic Instruction Code) para DOS. Mas as versões atuais dos dois produtos, Delphi 3.0 e Visual Basic 5.0, os deixaram muito parecidos. Esta nova versão do Visual Basic trouxe uma real orientação a objetos, e a geração de executáveis para o código nativo do processador. Segundo Santos (1997), escolher uma das linguagens como sendo a melhor, é uma tarefa que envolve muitos pontos de preferência pessoal. Basicamente, ambos podem executar as mesmas funções com pequenas variações na facilidade de implementação e

estilo de trabalho. De acordo com Santos (1997), o Delphi é mais flexível, enquanto o Visual Basic é mais prático. Optou-se pelo Visual Basic, que possui as seguintes características:

- recursos de acesso a dados, que permite criar bancos de dados e aplicativos para os formatos mais conhecidos de bancos de dados;
- tecnologia ActiveX, que permite usar a funcionalidade oferecida por outros aplicativos;
- recursos de Internet, o que proporciona acesso a documentos e aplicativos pela Internet;
- o aplicativo executável gerado, utiliza bibliotecas de vínculo dinâmico (DLL – Dynamic Link Library), o que permite organizar melhor o programa e criar executáveis menores.

Além dessas características, o Visual Basic é mais utilizado e difundido que o Delphi, pois o sistema de programação Applications Edition, que também utiliza esta linguagem, está incluído no pacote Microsoft Office, o conjunto de aplicativos mais vendido no mundo.

Neste capítulo foi apresentado o embasamento teórico para o desenvolvimento do sistema de supervisão que será apresentado no capítulo 5, e programas de controle e monitoração que serão apresentados no capítulo 4.

No próximo capítulo será vista uma descrição completa e detalhada de todos os elementos constituintes do projeto da Plataforma Industrial para Pesquisa, Ensino e Formação em Automação (PIPEFA);

Capítulo 3

Descrição do Projeto PIPEFA

Neste capítulo serão vistos todos os elementos constituintes da Plataforma Industrial para Pesquisa, Ensino e Formação em Automação – PIPEFA. Será apresentado o esquema de funcionamento de cada posto de trabalho, a comunicação entre eles e descrito o sistema de Gestão de Produção.

O projeto PIPEFA foi desenvolvido para o ensino, pesquisa e formação tecnológica na área de automação industrial. Foi elaborado por Rosário (1996), para ter baixo custo operacional, e capacidade de desenvolvimento de aplicações em automação, que possibilitem melhoria de desempenho, menores custos, maior flexibilidade e qualidade de um sistema de manufatura. A plataforma utiliza componentes de mercado, tais como: CLP, cilindros pneumáticos, sensores magnéticos e leitor de código de barras.

A plataforma realiza operações típicas de Sistemas Automatizados de Produção, como carregamento, descarregamento, transferência, execução de operações de montagem, planejamento, controle de qualidade, controle de processos e gestão de produção. O projeto PIPEFA estabeleceu um acordo de cooperação entre o Laboratório de Automação Integrada e Robótica da UNICAMP, o Instituto de Automação da Fundação CTI e o LIISI (Laboratoire d'Ingénierie Intégrée des Systèmes Industriels – França), como pode ser visto em Rosário (1999).

Na França, onde a plataforma já se encontra em operação há alguns anos, é denominada PEREGA (Plateforme Expérimentale pour la Recherche et l'Enseignement en Génie

Automatique). Este acordo de cooperação permitiu o aprofundamento de conhecimentos, realizações no ponto de vista material e sua aplicação em Sistemas Automatizados de Produção, dentro dos objetivos de geração de uma massa crítica de pesquisadores no domínio da Engenharia de Automação Integrada.

A Plataforma PIPEFA permite a montagem e o desmonte de cubos numa placa de base e um sistema de transferência acionado através de um motor elétrico cc. A partir de um sistema de codificação (atualmente código de barras), acessível ao Sistema de Supervisão através do leitor de códigos de barras, determinados tipos de produtos poderão ser montados nos postos de trabalho. Cada posto será controlado a partir de um CLP TSX-17, onde, o Sistema de Supervisão terá acesso a algumas informações (provenientes de entradas, saídas e variáveis auxiliares). Existirá também um posto de inspeção, que permitirá verificar a conformidade do produto.

Um braço robótico realizará a tarefa de carregamento de cubos no posto de trabalho, sendo esse sistema controlado através de um “hardware” dedicado, ligado ao computador. Esse hardware permite a entrada e saída de informações a partir de um barramento I/O. Entretanto a configuração atual, normalmente utilizada na maioria das aplicações para ensino e pesquisa, não permite a utilização direta desse barramento, sendo necessário o desenvolvimento de um aplicativo. A partir disso pretende-se interligar essas entradas/saídas direto no CLP, de modo que o Sistema de Supervisão utilizado possa controlar/monitorar essas informações.

Ao mesmo tempo, o fluxo das placas de base que transitam no sistema de transferência, e que será destinada a montagem/desmontagem de cubos, deverá ser controlada a partir do Sistema de Supervisão. Esse sistema permitirá a monitoração, gerenciamento de informações, tratamento gráfico e estatístico de dados, permitindo a automação completa do problema proposto.

Assim sendo, a integração de todos esses elementos, permitirá a capacitação do desenvolvimento de conhecimentos na área de Integração de Sistemas Automatizados, permitindo a extensão do problema aos demais postos constituintes da Plataforma PIPEFA e a

outras aplicações industriais. Para uma melhor compreensão das variáveis utilizadas é apresentado no Anexo II uma descrição completa das entradas e saídas do sistema.

3.1 Funcionamento do Sistema

A plataforma realizará a confecção de um produto genérico constituído de uma placa de base e cubos menores, do tipo *LEGO®* (como mostrado Figura 3.1), que além de apresentarem um baixo custo possuem boa precisão mecânica. Esse produto será realizado a partir da montagem, nas placas de base, de cubos em diferentes posições e em até dois níveis. Dependendo do número, posições e, futuramente, das cores dos cubos montados nessas placas, os produtos finais serão considerados diferentes.

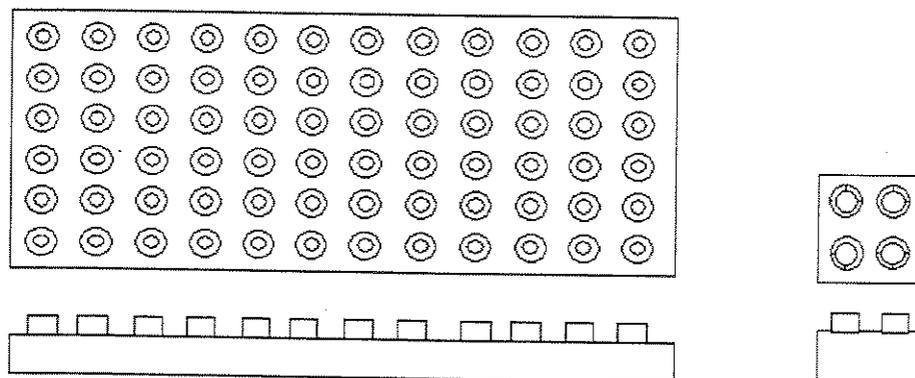


Figura 3.1: Placa Base e Placa de Montagem

O processo de produção se inicia com o recebimento de uma placa de base, inserida no **Sistema de Transferência pelo Posto de Carregamento**, sobre a qual podem ser encaixados cubos em três diferentes posições e em até dois níveis. A colocação dos cubos é feita nos dois **Postos de Montagem** (designados **Central** e **Lateral**), totalizando uma família de 26 produtos diferentes (a lista completa de produtos encontra-se no Anexo I). Os Postos de Montagem são alimentados por meio de um **Robô Industrial**, que neles insere cubos. Este robô poderá atuar também como um sistema para armazenagem intermediária.

Em seguida, o **Posto de Inspeção** recebe as placas montadas e verifica a montagem feita. Caso o produto seja aprovado, o mesmo deverá ir para o **Posto de Descarregamento** para ser estocado. No caso de haver um defeito de montagem, esses produtos poderão ser rejeitados ou desmontados nos **Postos de Desmontagem** (os mesmos: **Central e Lateral**), para retornar ao sistema sob a forma de matéria-prima (cubos e placas individuais).

Essas diferentes formas de montagem possibilitarão a utilização do conceito de Tecnologia de Grupo, a ser explorado através de um sistema de gestão de produção, no qual metodologias e algoritmos poderão ajudar na determinação da melhor seqüência de fabricação em função da capacidade de máquinas, prazo de entrega, lucro, etc.

A parte operacional e uma parte do sistema de supervisão cooperativa e comando estão sendo instalados no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP, Instituto de Automação da Fundação CTI e o LIISI da França. A longa distância que separa esses centros de pesquisa permitirá a utilização do conceito de empresa estendida, a integração de dois diferentes grupos de pesquisadores e, principalmente, a realização de um trabalho na área de automação industrial em forma integral, que considere desde aspectos ligados a concepção de um produto e implantação de um sistema automatizado de manufatura até sua integração com os níveis de gerenciamento de produção.

Finalmente, esta plataforma deverá considerar todos os níveis da "pirâmide CIM" (gestão global, gestão da produção, controle de processos e "chão de fábrica"). Assim, poderá servir de suporte aos estudos e pesquisa na integração das diferentes atividades de uma empresa.

Atualmente, a plataforma retrata totalmente um sistema de produção discreta, sendo constituída dos seguintes postos de trabalho:

- Sistema de Transferência;
- Carregamento de placas de base;
- Leitura de Código do Produto;

- Montagem/Desmontagem Central;
- Montagem/Desmontagem Lateral;
- Inspeção de um Produto;
- Descarregamento com sistema de armazenagem
- Manipulador robótico para reposição de produtos

Cada um desses postos de trabalho é constituído de uma Parte de Comando e uma Parte Operativa (vide Figura 3.2). A Parte de Comando consiste em um CLP onde será realizado o processamento lógico de sinais de entrada e saída. A Parte Operativa consiste nos motores, atuadores elétricos, pneumáticos, hidráulicos, e todos outros elementos físicos que participam da manipulação e processamento dos produtos.

Os postos constituintes deste Sistema Automatizado de Produção (SAP) executarão suas respectivas operações de maneira totalmente realista com relação à Parte de Comando, visto que CLPs são os equipamentos de controle comumente utilizados no meio industrial para este fim. Ao mesmo tempo, seu projeto é suficientemente próximo a realidade com relação a Parte Operativa de um SAP tradicional, de forma a validar uma arquitetura que permita um diálogo concreto com o meio industrial. A seguir, os postos de trabalho serão analisados com mais detalhes.

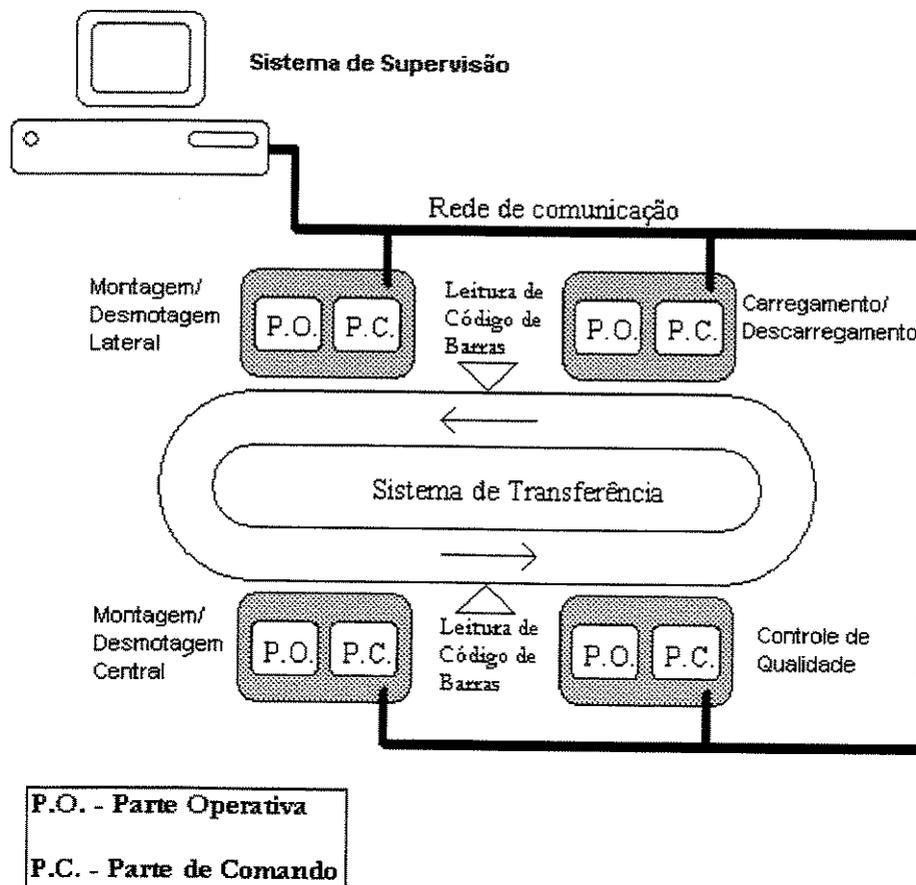


Figura 3.2: Layout da Plataforma PIPEFA

3.1.1 Posto de Montagem/Desmontagem de Produtos

Dois postos realizam a função de transformação de forma e características das peças através de operações de montagem e desmontagem dos cubos *LEGO*® nas placas de base, em função do plano de produção.

Atualmente as placas são identificadas por um código de barras fixado em cada uma delas. Sobre elas são montados cubos em três posições (lateral direita, central e lateral esquerda) e em dois níveis de altura. Com o objetivo de simular um sistema multiproduto e de criar um elenco

de seqüências de montagem ou desmontagem de produtos, foi construído um posto para realização de operações de montagem/ desmontagem de cubos na **posição central** e outro para montagem/desmontagem de cubos na **posição direita e/ou esquerda (lateral)**. Este último tem operação idêntica ao primeiro, mas seu sistema de transferência dispõe de um cilindro giratório que se encarrega de dar uma rotação de 180° na placa de base. Uma vez completado o giro, este posto deve apenas realizar a mesma operação, com a diferença de que esta acontece no lado oposto.

3.1.2 Posto de Carregamento de Placas de Base

Este posto de trabalho, é constituído do carregamento de placas de base no sistema de transferência. As placas de base são introduzidas no sistema de transferência, de acordo com as ordens de produção previamente estabelecidas pelo Sistema de Gestão de Produção (gerenciadas a partir do Sistema de Supervisão). Um conjunto de placas de base codificadas ficam disponíveis num sistema de armazenagem, esperando uma ordem para entrada no sistema de transferência.

3.1.3 Posto de Descarregamento de Produtos

Após um determinado produto passar pelo Posto de Inspeção, levando-se em conta sua inspeção, o mesmo poderá ser descarregado e armazenado em estoque (produto final) ou ser retirado temporariamente do Sistema de Transferência.

3.1.4 Posto de Inspeção

O Posto de Inspeção verificará a conformidade do produto, em relação a quantidade e posicionamento dos cubos montados. A partir do mapeamento do produto realizado através de sensores de fibra ótica, ele fornecerá ao Sistema de Supervisão informações do produto realizado.

Além disso o Sistema de Supervisão poderá decidir se um produto segue os padrões estabelecidos pelo Sistema de Informação e Gestão de Produção a respeito da qualidade do mesmo quanto ao tipo de montagem mecânica. Isto é verificado de duas formas distintas.

- Num primeiro nível de exigência, através de transdutores de fibra ótica dispostos em duas alturas, que serão utilizados para detecção da presença de cubos, montagens incorretas, etc. Esta detecção levará em conta tão somente a presença de cubos em cada uma das seis posições possíveis.
- Num segundo nível de exigência, também as cores de cada cubo serão determinantes para a diferenciação de produtos. Neste caso, um sensor de nível de cinza verificará as cores de cada cubo (e, em consequência, também sua presença). Mais precisamente, cada nível de cinza é transformado em um valor de tensão elétrica proporcional (logo, sinal analógico); através de uma interface analógico-digital presente no Posto de Inspeção, o Sistema de Supervisão estará relacionando cada uma das cores básicas de cubos a **faixas de tensão** bem definidas.

Este tratamento de sinais fará com que a PIPEFA trabalhe tanto com variáveis discretas como analógicas.

A hierarquia de exigência será definida e acessada através do Sistema de Supervisão e esta, em conjunto com o resultado obtido pela verificação ocorrida no Posto de Inspeção, definirá se o produto foi adequadamente montado.

As informações obtidas neste posto são muito importantes para a Supervisão e Gestão da Produção que deverá ser implementado no Instituto de Automação da Fundação CTI. O Sistema de Supervisão, com base em estatísticas dinâmicas, pode saber se um posto de montagem apresentou bom desempenho ou não, e atuar no sentido de corrigir distorções (em caso de níveis de qualidade abaixo do desejado).

3.1.5 Sistema de Transferência

Cada posto de trabalho possui um Sistema de Transferência próprio, constituído de uma esteira acionada por um motor de corrente contínua. Entretanto, a Parte Comando está centralizada através de um Controlador Lógico Programável, e é responsável pela integração mecânica dos postos existentes na célula. O Sistema de Supervisão, baseado nas informações de cada produto (etiquetado a partir de código de barras), tomará decisões de parada, montagem e desmontagem, carregamento e descarregamento de placas, baseados num plano de produção.

3.1.6 Leitura do Código de um Produto

Após o Posto de Carregamento de Placas de Base, existe um Leitor de Código de Barras, ligado fisicamente ao teclado do computador que está executando o programa de supervisão. Após a leitura do código de barras, o Sistema de Supervisão associa essa informação ao tipo de produto que será realizado nas próximas operações através dos postos de Montagem/Desmontagem Central e Lateral.

3.1.7 Manipulador Robótico

A partir de possíveis ordens de um Sistema de Supervisão e da detecção de falta de cubos no sistema de armazenamento dos Postos de Montagem Central e Lateral (feitas através dos sensores de nível de cubos em estoque neles presentes), manipuladores robóticos ROBIXTM disponíveis na UNICAMP colocam cubos nestes postos de trabalho até completar o estoque. O mesmo robô poderá atuar também como um sistema para armazenagem intermediária de produtos.

3.1.8 Sistema de Supervisão

A integração entre os postos de trabalho e a gestão de produção é realizada a partir de um Sistema de Supervisão que tem como uma de suas tarefas receber os dados do “**chão de**

fábrica” e colocá-los a disposição dos **níveis superiores de gerenciamento**. Esta integração possibilita o acompanhamento em tempo real de variáveis e estados representativos das operações em curso no chão da fábrica (postos de trabalho), com a finalidade de tomada de decisões de operação, otimização dos processos e criação de registros históricos.

A supervisão atua de maneira automática e normalmente conta com o auxílio de um operador, que poderá interferir no sistema através de uma interface. Este é um elemento de fundamental importância no Sistema de Supervisão, devendo permitir a monitoração dos processos de modo hierárquico.

A aquisição de dados provenientes dos **Postos de Trabalho** a serem repassados ao Sistema de Supervisão, é realizada através de uma interface serial, rede UNITELWAY ou FIPWAY se comunicando com os Controladores Lógicos Programáveis (TSX-17 e TSX-37).

Este sistema recebe também orientações do **Sistema de Gestão da Produção** para determinar as operações de produção. Assim sendo, o Sistema de Supervisão deve dialogar com os sistemas localizados **hierarquicamente acima e abaixo dele** (Figura 3.3).

O Sistema de Supervisão deve ser capaz de se comunicar com as Partes de Comando dos postos (seus respectivos CLPs), além de oferecer uma interface adequada a operadores, tanto no aspecto de acompanhamento de processo (visualização, alarmes, etc) como no de controle. Atualmente, além do sistema de supervisão desenvolvido neste trabalho, estamos avaliando dois Sistemas Supervisores Industriais disponíveis na UNICAMP (WIZCONTM e INTERACTTM).

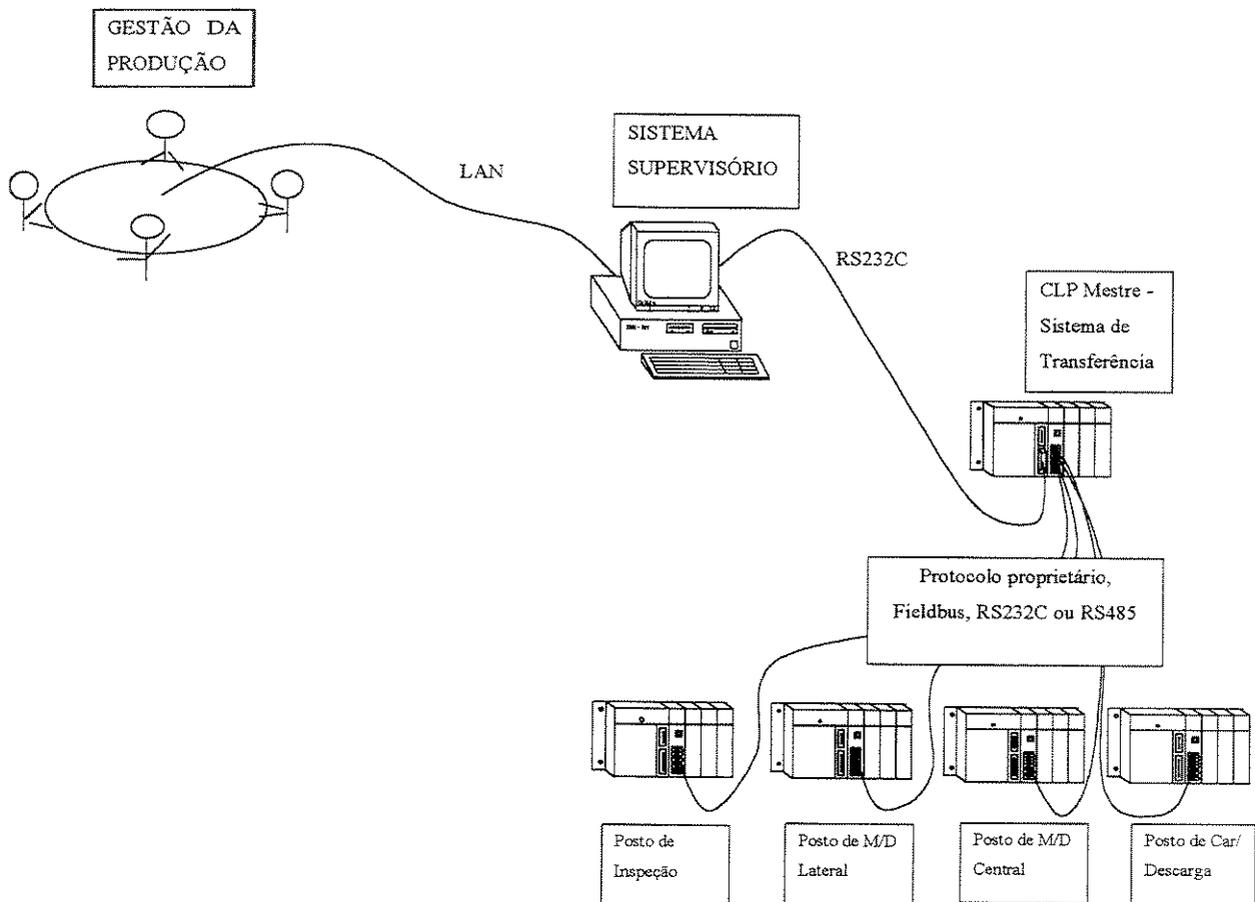


Figura 3.3: Hierarquia de Funcionamento do Projeto PIPEFA

3.2 Descrição da Comunicação entre os Postos de Trabalho

O Sistema de Supervisão além de controlar a comunicação entre os postos de trabalho, também serve de elo de ligação entre a parte de *software* e *hardware*, como mostrado na Figura 3.4.

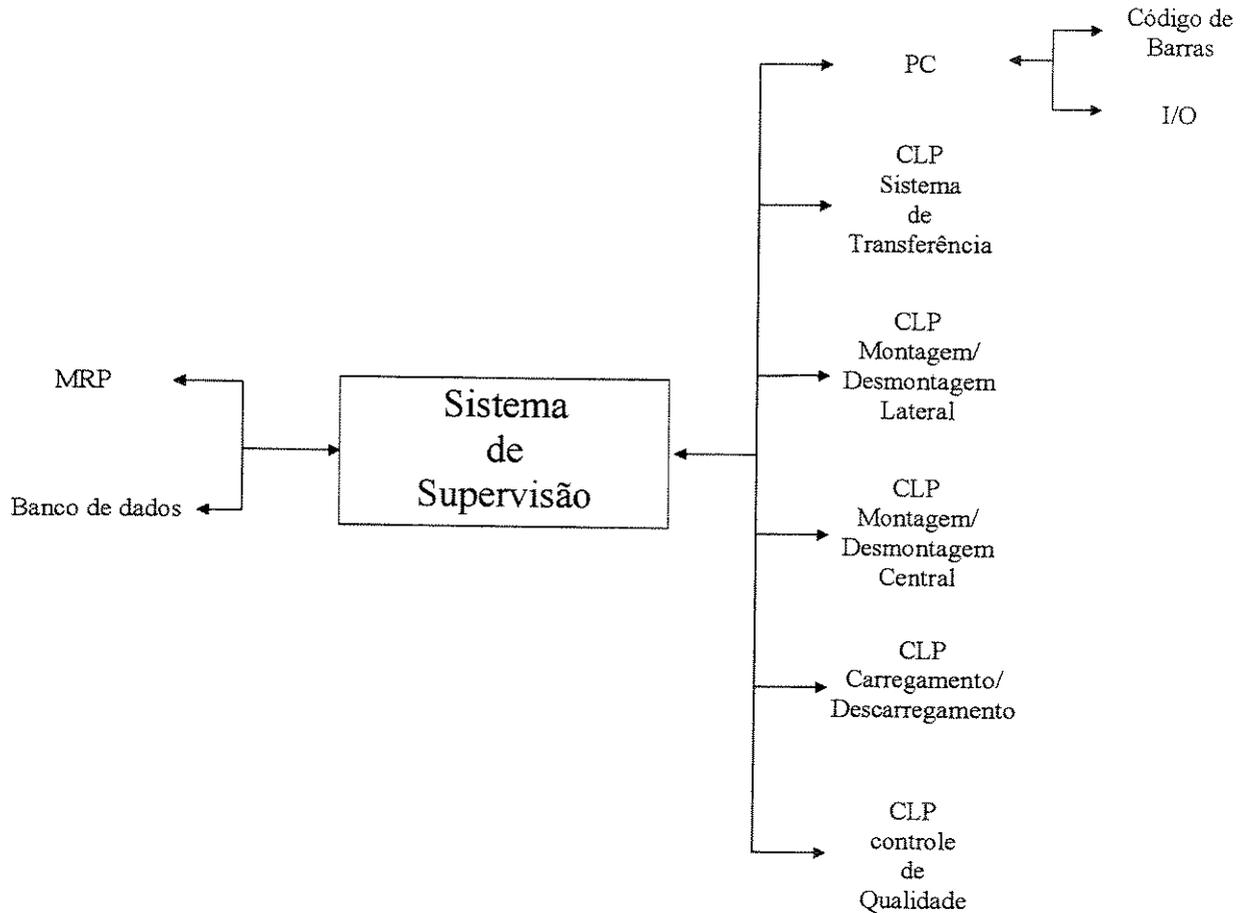


Figura 3.4: Representação esquemática do papel do sistema de Supervisão

A seguir será explicado como acontece a comunicação entre os postos de trabalho.

- Código de Barras I

Um leitor de código de barras está conectado ao Sistema de Supervisão. O leitor tem a finalidade de ler o código de barras fixado nas placas de base antes do posto de montagem/desmontagem central. O leitor utilizará a interface do teclado do microcomputador para a conexão com o sistema de supervisão.

- Banco de Dados

De posse do código de barras, o Sistema de Supervisão o utiliza para associá-lo, no banco de dados, com o produto que será montado (código de montagem). Esta associação pode ser editada e refeita à qualquer momento, mesmo externamente, como por exemplo pelo Sistema de Gestão de Produção, possibilitando uma programação dinâmica e flexível de produção.

Além de conter o código de montagem e o de barras, o Banco de Dados armazenará também dados referentes a qualidade na montagem dos produtos especificados (aferida pelo Posto de Controle de Qualidade) e a outros aspectos de produção (tempos de processamento de cada produto, interrupções, etc.). A composição estatística dessas informações será extremamente útil para determinar os parâmetros de confiabilidade dos processos da PIPEFA.

A partir da definição de tipo de produto, proveniente do Banco de Dados, o Sistema de Supervisão está apto a converter esta especificação para as informações que o CLP de cada posto de trabalho deve receber.

- Sistema de Transferência

O CLP do Sistema de Transferência deve ser informado se o produto deverá utilizar o Posto de Montagem/Desmontagem Central e o Posto de Montagem/Desmontagem Lateral e, caso este último venha a ser utilizado, se haverá processamento de cubos do lado direito, esquerdo ou de ambos (a partir desta última informação, o cilindro giratório do Posto M/D Lateral poderá ou não sofrer uma rotação de 180°, a partir de seu lado natural, *default*). Desta forma, três informações serão necessárias. Em formato binário, 3 *bits*, cada um deles com 1 ou 0 (sim ou não), como mostrado na Figura 3.5.

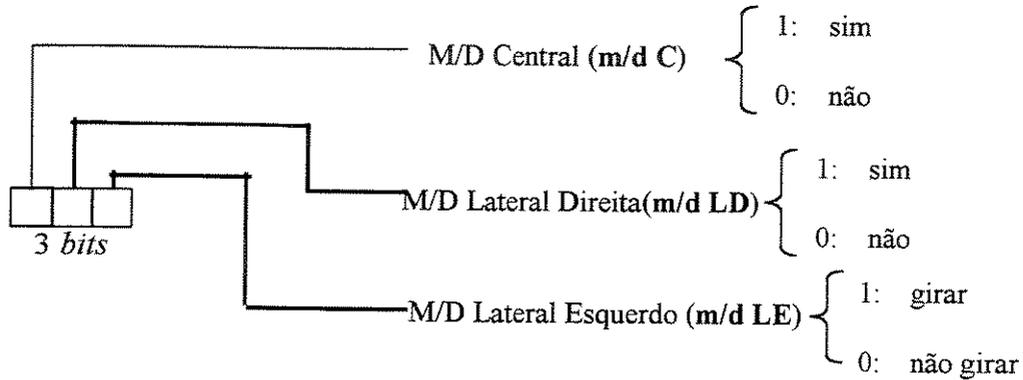


Figura 3.5: Informações para o Sistema de Transferência

- Posto de Montagem/Desmontagem Central

Este posto monta ou desmonta o(s) cubo(s) central(centrais) na placa de base. As informações para este posto são: se deve haver montagem ou desmontagem e o número de cubos: 1 ou 2 cubos. Observe que a possibilidade de 0 cubos (nenhum) já foi especificada pela informação passada ao Sistema de Transferência (M/D Central: 0 ou 1). Logo, necessitamos de duas informações, ou 2 *bits*, como mostrado na Figura 3.6.

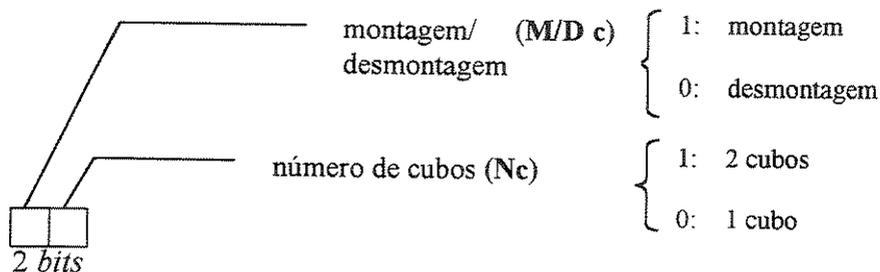


Figura 3.6: Informações para o Posto de Montagem/Desmontagem Central

- Posto de Montagem/Desmontagem Lateral

Analogamente ao posto anterior, as informações são: montagem ou desmontagem e 1 ou 2 cubos, informações estas que deverão ser especificadas para ambos os lados: direito e esquerdo, perfazendo um total de 4 *bits*, como mostrado na Figura 3.7.

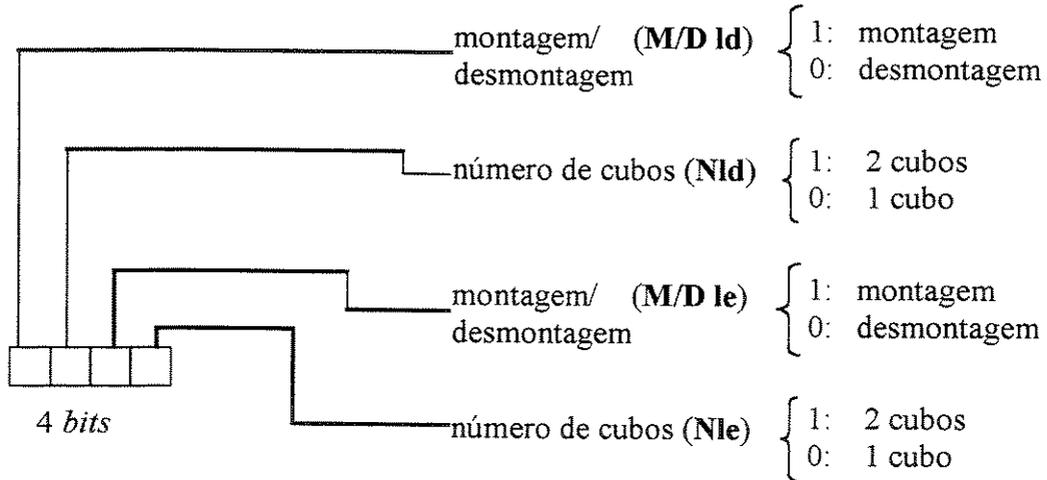


Figura 3.7: Informações para o Posto de Montagem/Desmontagem Lateral

- Código de Barras II

Depois de passados os dois postos de trabalho em que há manipulação das unidades constituintes dos produtos, os cubos *LEGO*®, a placa se dirige ao Posto de Controle de Qualidade. Antes dele, porém, há uma segunda leitura de seu código de barras (Código de Barras II). Os motivos para essa segunda leitura, devem-se ao fato de estar previsto uma implementação futura onde a seqüência de montagem das bases poderia ser alterada. Sendo assim, para se garantir qual base estaria entrando no posto de controle de qualidade, necessita-se de nova leitura do código de barras.

- Posto de Controle de Qualidade

Já dentro deste posto, transdutores de fibra óptica detectam a presença de cada um dos cubos. Essa informação é decomposta em sinais elétricos: 1 ou 0 (presença ou não presença de cubos). Como o número máximo possível de cubos é 6, a CLP do Posto de Controle de Qualidade enviará seis *bits* ao Sistema de Supervisão. O formato desta palavra de seis *bits* é mostrado na Figura 3.8.

O Sistema de Supervisão consulta mais uma vez o Banco de Dados, mais especificamente sua lista de equivalência entre código de barras e código de montagem do produtos. O código de montagem que equivale ao código de barras lido logo antes do Posto de Controle de Qualidade - a disposição de seus cubos - é comparado com esses 6 *bits*, um a um. Essa comparação caracteriza o controle de qualidade dos processos da PIPEFA: se existe equivalência entre o produto efetivamente montado (os cubos detectados, cada um dos 6 *bits*, pelo Posto de Controle de Qualidade) e o produto especificado pelo plano de produção (código de montagem, da lista de equivalência do Banco de Dados). Surgem duas possibilidades:

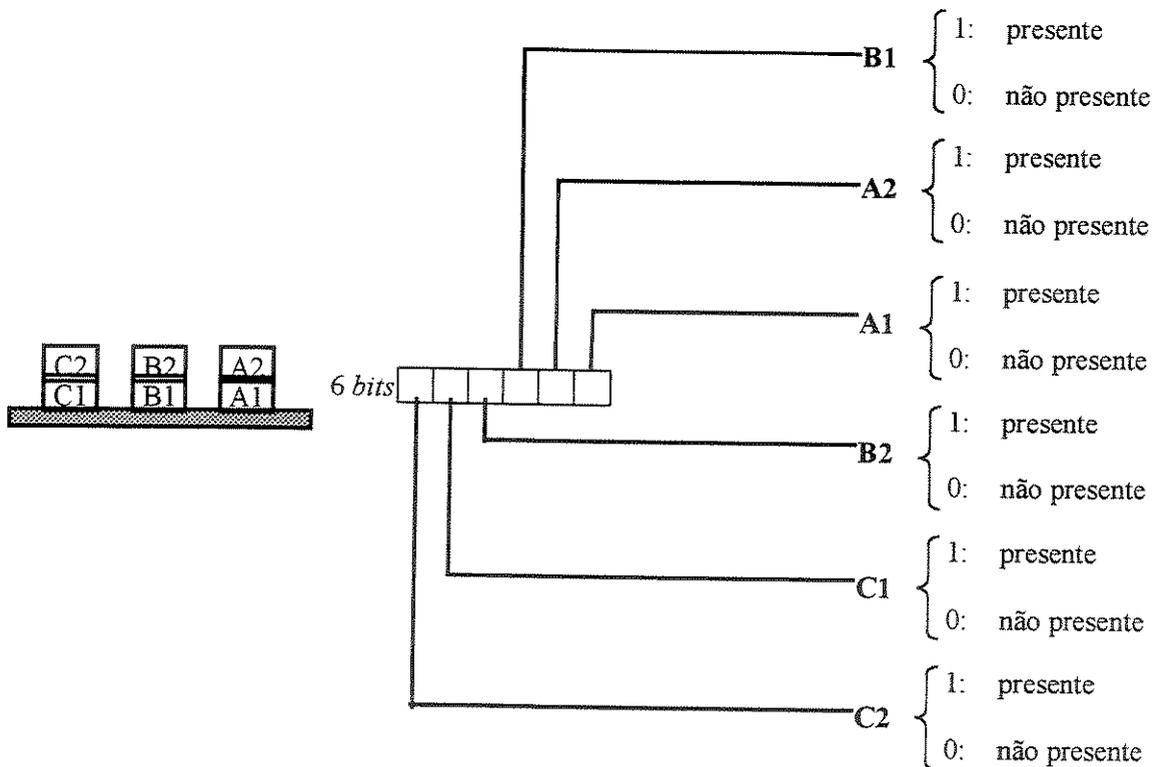


Figura 3.8: Informações do Posto de Controle de Qualidade

1. a total compatibilidade entre um e outro, o que indica que as operações do Sistema de Transferência e Postos de Montagem/Desmontagem estão de acordo com o plano de produção e não houve defeitos de montagem; neste caso, pode-se dar continuidade ao processamento dos demais produtos de uma lista de produção (as decisões sobre seqüência de produtos, prioridade de montagem, etc., é feita pelo Sistema de Gestão de Produção, que será descrito mais adiante);
2. caso não tenha havido essa equivalência, a disparidade de presença/posição de **cada um dos cubos defeituosos** é totalmente conhecida; a partir do complemento lógico do produto defeituoso em relação ao produto perfeito, o Sistema de Supervisão pode (se esta for a prioridade estabelecida pelo Sistema de Gestão de Produção), controlar todo um novo processo de montagem dos cubos faltantes e desmontagem de cubos em excesso.

Convém lembrar que esta segunda situação, de não conformidade entre produto fabricado e aquele planejado, pode vir a ocorrer não necessariamente por causa de falhas nos postos de trabalho (falha de atuadores, travamento, etc), mas também pelo caráter dinâmico do planejamento de produção da plataforma: como dito anteriormente, o Banco de Dados pode ser mudado a qualquer momento pelo Sistema de Gestão de Produção. Esse é mais um dos aspectos em que a PIPEFA simula um Sistema Automatizado de Produção real: a flexibilidade que o meio industrial deve ter para modificar processos em tempo de execução.

Desta forma, no meio do processo, o produto que está sendo montado pode deixar de ser prioritário e outro diferente pode ser associado àquele código de barras, pela simples adição ou substituição de arquivos do Banco de Dados, em tempo real. E, é claro, a comparação dos dados a partir do Posto de Controle de Qualidade não estaria em conformidade com o novo produto desejado.

Finalmente, no caso de produtos não aprovados, o Sistema de Supervisão deve seguir critérios da Gestão de Produção, a fim de decidir se eles devem ser eliminados como refugos ou se procede nova montagem ou desmonte.

- Posto de Carregamento/Descarregamento

O Sistema de Supervisão pode então, a partir do controle de qualidade realizado no posto anterior, recolher as placas já montadas, descartar as placas defeituosas e carregar a esteira do Sistema de Transferência com placas novas, para iniciar um novo processo. Tem-se então, 3 informações diferentes (3 *bits*), como mostrado na Figura 3.9.

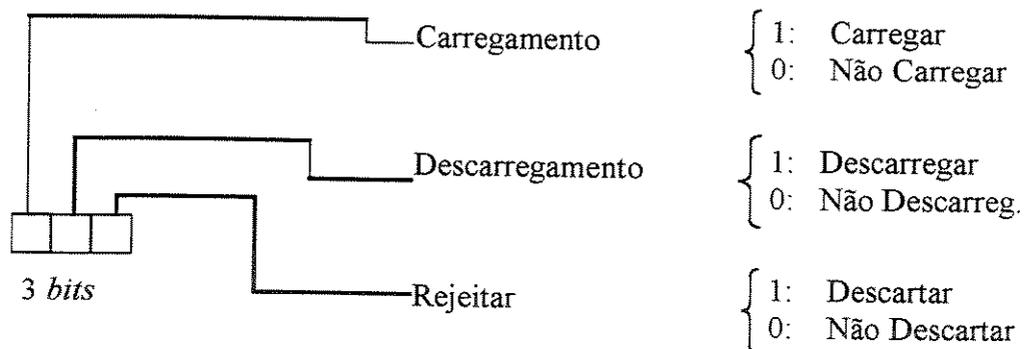


Figura 3.9: Informação para o Posto de Carregamento/Descarregamento

3.3 Sistema de Gestão de Produção

A função do Sistema de Gestão de Produção é fornecer um plano de produção para os diferentes postos constituintes do SAP em desenvolvimento. Procurando dotá-lo dos requisitos de um sistema real, ele trabalha numa estrutura com vários níveis, onde a decisão de mais longo prazo torna-se a entrada e a meta para o próximo nível de decisão.

Atualmente, uma parte das atribuições do Sistema de Gestão de Produção estão inseridos no Sistema de Supervisão, em linguagem do VISUAL BASIC . Nesse estágio, o programa lida com a hierarquia desejada no critério de aceitar, rejeitar ou remontar produtos “defeituosos”. O procedimento estaria estruturado em três níveis:

Nível 0 - qualquer produto não adequado é descarregado imediatamente como refugo

Nível 1 - produtos não adequados são descarregados e armazenados temporariamente até que a plataforma tenha terminado seu plano de produção; a seguir são carregados no sistema de transferência para remonte

Nível 2 - produtos não adequados retornam imediatamente para remonte

Essas prioridades podem ser alteradas em tempo real através do sistema de Supervisão.

Na configuração idealizada para este trabalho, o Sistema de Supervisão estará fisicamente localizado no LAR e o Sistema de Gestão da Produção no Instituto de Automação da Fundação CTI. A interligação das entidades existe via Internet.

Neste capítulo foram vistos todos os elementos constituintes da Plataforma Industrial para Pesquisa, Ensino e Formação em Automação – PIPEFA. Foram apresentados o esquema de funcionamento de cada posto de trabalho, a comunicação entre eles e descrito o sistema de Gestão de Produção.

No próximo capítulo serão vistos sistemas de supervisão aplicados a sistemas automatizados de produção, no qual serão relatados os experimentos realizados que deram subsídios ao desenvolvimento do sistema de supervisão aplicado ao projeto PIPEFA.

Capítulo 4

Utilização de Sistemas de Supervisão em Automação

Neste capítulo serão apresentadas três aplicações de sistemas de supervisão à sistemas automatizados de produção, desenvolvidas e implementadas no LAR. São apresentadas diversas ferramentas de desenvolvimento, e expostas suas qualidades e limitações. São discutidas as dificuldades encontradas na utilização do ambiente de programação adotado, e mostradas as soluções encontradas.

4.1 Sistema de Supervisão

Um Sistema de Supervisão é, a rigor, um sistema de comunicação no sentido mais amplo da palavra, pois engloba:

- visualização de processos;
- sistema de informação;
- interface com usuário no que se refere à intervenção de controle;
- ligação entre a parte operacional dos processos e os sistemas mais altos na hierarquia de planejamento.

Atualmente um Sistema de Supervisão é muito utilizado em manufatura, nos processos automatizados em que a integração entre os diferentes postos de trabalho e a gestão de produção

é realizada a partir do Sistema de Supervisão, que tem como uma das tarefas receber os dados do “chão de fábrica” e colocá-los a disposição dos níveis superiores de gerenciamento. Esta integração possibilita o acompanhamento em tempo real de variáveis e estados representativos das operações em curso no “chão da fábrica” (postos de trabalho), com a finalidade de tomada de decisões operacionais, otimização dos processos e criação de históricos.

A aquisição de dados provenientes dos postos de trabalho, a serem repassados ao Sistema de Supervisão, é realizado por CLPs e computadores com interfaces de aquisição de informações (código de barras, *Input/Output*, etc.). Este sistema recebe também orientações do Sistema de Gestão da Produção para determinar as operações de produção. Logo, o Sistema de Supervisão deve dialogar com os sistemas localizados hierarquicamente acima e abaixo dele. Convém ressaltar que todas essas funções devem ser desempenhadas em tempo real, para que a monitoração e controle de informações funcione adequadamente.

A utilização de um software especialmente desenvolvido para a supervisão de sistemas, como o P-CIM, WIZCON, INTERACT, nem sempre é tão prático como se imagina inicialmente. Primeiramente este tipo de software é muito caro e só serve para a supervisão de sistemas; depois ele trabalha com uma linguagem de programação com a qual o programador deverá se habituar; e finalmente, quando se esbarra com alguma limitação do sistema, como por exemplo um “drive” de comunicação, fica-se na dependência do fabricante resolver o problema, pois estes *software* são pacotes fechados, limitados para implementação de mudanças.

A utilização do Visual Basic para desenvolvimento do sistema de supervisão aumenta a interação do sistema com aplicativos do Windows®, aumenta a flexibilidade (pois existem vários programadores e construtores de bibliotecas) e diminui os custos de aquisição e instalação. Os fabricantes de CLPs apostam nesta interação com aplicativos do Windows®, tanto é que começam a surgir aplicativos que permitem Leitura e Escrita de dados diretamente na memória do CLP, como o DDE Server, que utiliza os recursos do protocolo DDE (Dynamic Data Exchange), para fazer um intercâmbio dinâmico de dados, entre programas baseados no Windows®, com o CLP da Koyo.

4.2 Implementação de um Programa de Controle para Transporte de Peças

Para melhor verificar as características de cada uma das linguagens de programação de CLPs, utilizou-se o software Visual Basic para programar um experimento em duas linguagens, em Grafset e em Linguagem de Relés. O experimento consiste em automatizar uma estação de montagem de cubos Legos®, que monta os cubos em três posições distintas da base de montagem.

As bases de montagem são colocadas na esteira número um, que tem por função abastecer a esteira número dois, evitando amontoamento das bases. A esteira número dois, transporta as bases até o elevador. O elevador transporta as bases para o dispositivo de montagem, controlado pelo CLP. Após a montagem, o elevador então retorna a base montada para a esteira número dois, que a transporta para a esteira seguinte. Uma representação do funcionamento do sistema é mostrado na Figura 4.1.

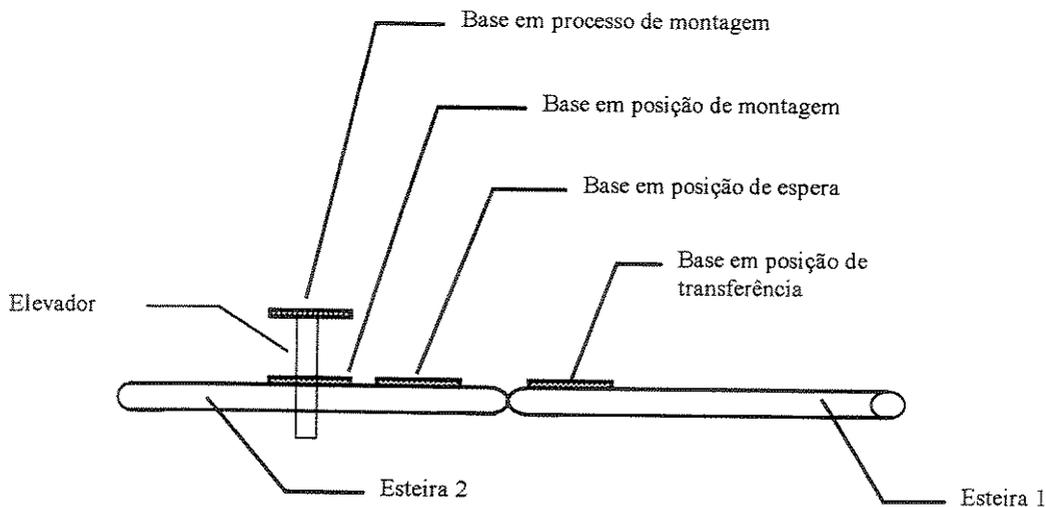


Figura 4.1: Representação do Sistema de Transporte de peças

O CLP utilizado para fazer a montagem dos cubos Legos® na base de montagem possuía poucas entradas e saídas, o que resultou na necessidade da utilização de outro equipamento para se fazer o controle das duas esteiras e do elevador, no caso um computador 386 DX 40.

Primeiro programou-se o processo de automação do transporte de bases em Diagrama de Relés, e depois reprogramou-se o mesmo processo em Grafcet.

4.2.1 Programação do Transporte de Peças em Diagrama de Relés

Para a programação da automação do transporte de bases em Diagrama de Relés, inicialmente montou-se o Caderno de Especificações, e a partir dele, a Tabela de Estados, com os estados dos sensores e atuadores para cada etapa do processo. A designação das variáveis envolvidas no processo é mostrado na Tabela 4.1. Desenvolveu-se um programa em Visual Basic capaz de ler o estado dos sensores, comparar estes estados com os da Tabela de Estados, e com isto controlar os atuadores. Dessa forma, conseguiu-se controlar três processos simultaneamente.

O primeiro processo a ser controlado é o controle do motor da esteira 1; o segundo processo é o controle da trava (pneumática) 1; o terceiro processo é o controle do elevador (cilindro pneumático). Uma representação de todo o sistema é mostrada na Figura 4.2.

O primeiro caso de estudo (controle do motor da esteira 1), consiste em controlar duas esteiras transportadoras e um elevador. A primeira esteira serve como uma espécie de magazine, fornecendo as bases de montagens para a esteira número dois. O primeiro processo a ser controlado pelo programa consiste em desligar a primeira esteira, toda vez que houver base de montagem em duas posições:

1. prestes a ser transferida para a esteira 2 (base de montagem no final da esteira número 1).
2. em posição de espera (base aguardando liberação do elevador para processo de montagem);

Tabela 4.1: Variáveis de Entrada e Saída para o Transporte de Peças

Variáveis de Entrada		Variáveis de Saída	
Variável	Significado	Variável	Significado
Sensor 1	Sensor Magnético 1	Stop 1	Trava de parada 1
Sensor 2	Sensor Magnético 2	Stop 2	Trava de parada 2
Sensor 3	Sensor Magnético 3	Motor 1	Motor da esteira 1
SER	Sensor Elev. Recuado	Motor 2	Motor da esteira 2
SEA	Sensor Elev. Atuado	Elevador	Pistão do elevador
SGF	Sensor Gav. Fechada	CLPout	Avisar CLP p/ início de mont.
SGA	Sensor Gaveta Aberta	FM	Variável Fim de Montagem
CLPin	CLP solicita desmont.		

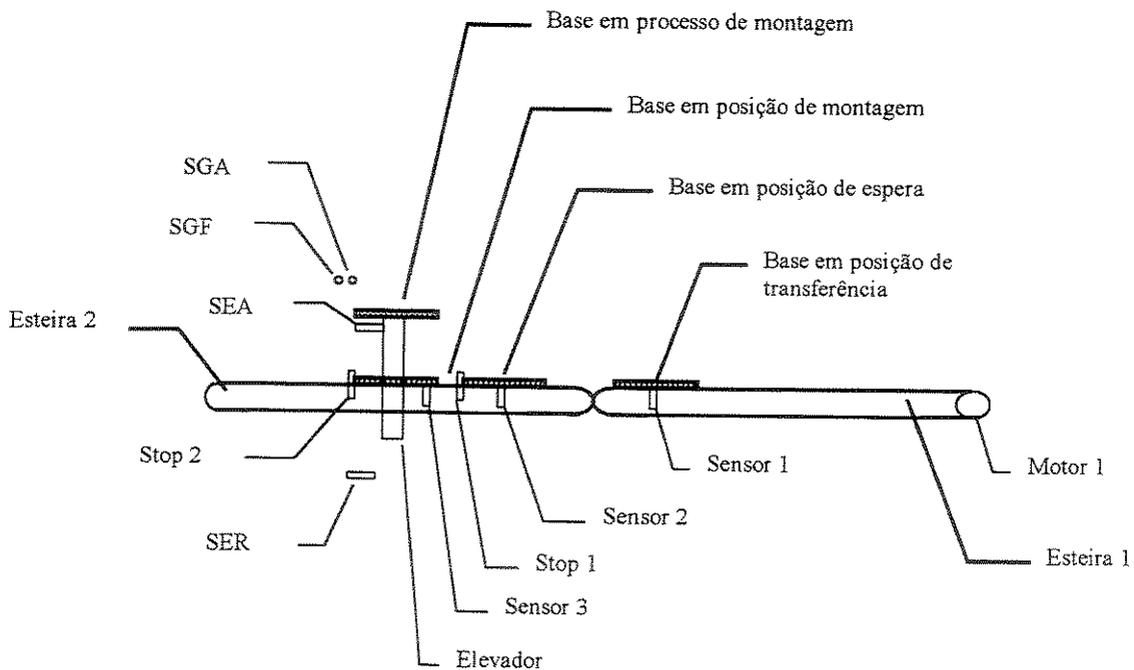


Figura 4.2: Variáveis do Sistema de Transporte de Peças

A Figura 4.3 facilita a criação do Caderno de Especificações do primeiro processo (controle do motor da esteira 1), o qual é descrito a seguir:

1. motor 1 permanece ligado sempre que não receber os sinais dos sensores 1 e 2 simultaneamente;
2. Quando receber os sinais dos sensores 1 e 2 simultaneamente, o motor 1 deve ser desligado.

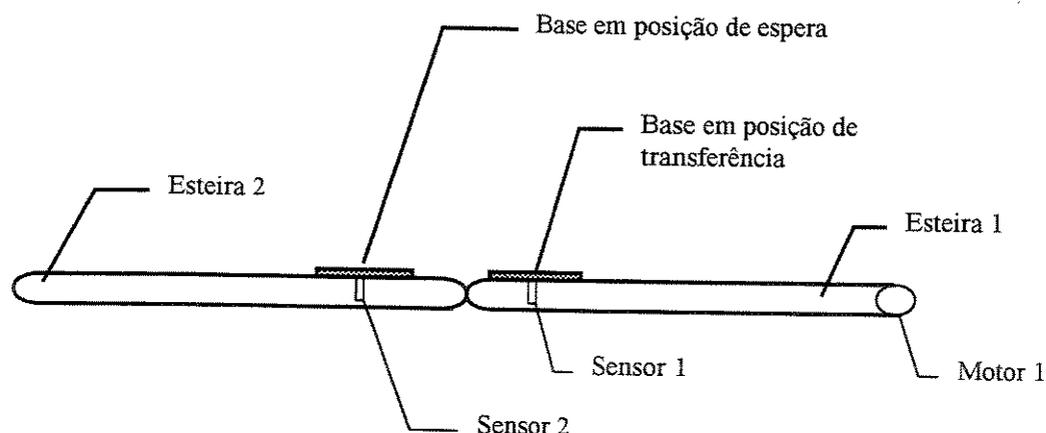


Figura 4.3: Posições das Bases para Desligamento da Esteira 1

De posse do Caderno de Especificações, pode-se agora construir a Tabela de Estados para o Controle do Motor da Esteira 1, a qual é mostrada a seguir:

Tabela 4.2: Tabela de Estados para o Controle do Motor da Esteira 1

Etapas	Sensor 1	Sensor 2	Motor 1
1	0	0	1
1	0	1	1
1	1	0	1
2	1	1	0

A partir da Tabela de Estados, pode-se utilizar a Linguagem de Relés, para se construir o programa de Controle do Motor da Esteira 1, mostrado a seguir:

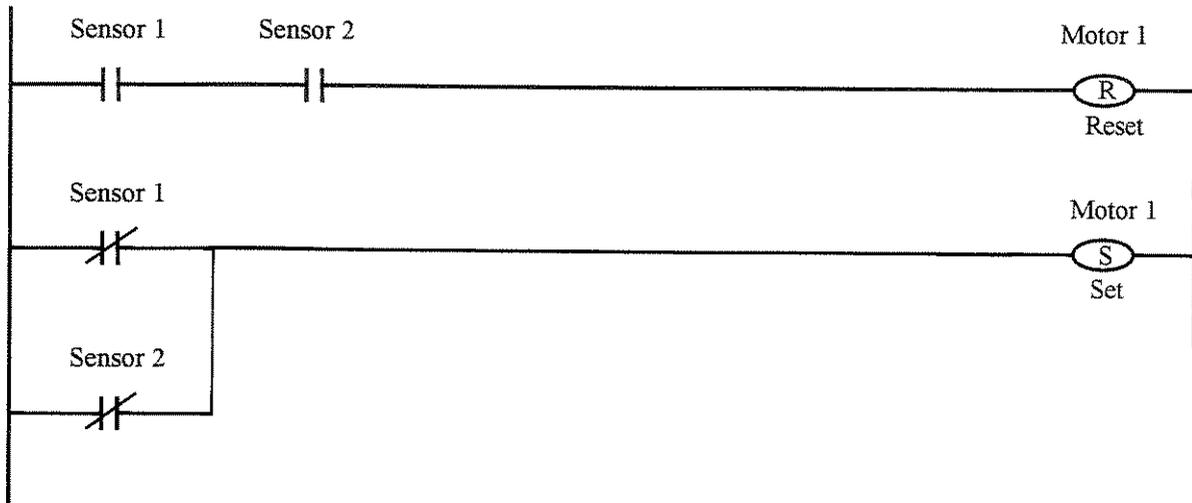


Figura 4.4: Diagrama de Relés do Controle do Motor da Esteira 1

O segundo processo a ser executado é o controle da trava 1. A segunda esteira serve para carregar/descarregar o elevador com as bases de montagem. O motor da segunda esteira permanece sempre ligado, pois ela possui uma trava que impede que a peça, em posição de espera, avance para a posição de montagem em um momento incorreto. A trava 1 (Stop 1) deverá ser desativada apenas quando não tiver base em montagem e tiver base em posição de espera. Novamente o primeiro passo é a construção do Caderno de Especificação. Ele foi montado baseando-se na Figura 4.5.

Caderno de Especificações para o segundo processo (controle da trava 1):

1. stop 1 permanece ativo até receber sinal do sensor 2, desde que não receba sinais da CLP (CLPin e CLPout);
2. Quando receber o sinal do sensor 2 e não receber sinais da CLP (CLPin e CLPout) o stop 1 deverá ser desativado por 3 segundos, para permitir a passagem da base.

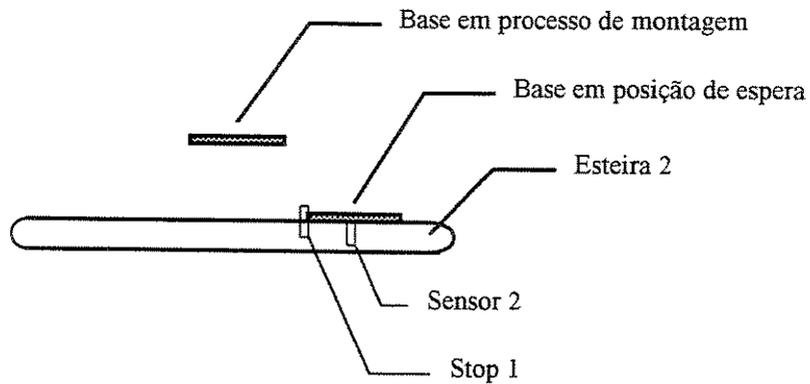


Figura 4.5 : Posições das Bases em Relação a Trava 1

Depois de montado o Caderno de Especificações pode-se criar a Tabela de Estados, mostrada a seguir:

Tabela 4.3: Tabela de Estados para o Controle da Trava 1 (Stop 1)

Etapas	Sensor 2	CLPin	CLPout	Stop 1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1
2	1	0	0	0
erro	1	1	1	1
erro	0	1	1	1

Finalmente pode-se construir o programa, em Linguagem de Relés, para Controle da Trava 1, mostrado na Figura 4.6:

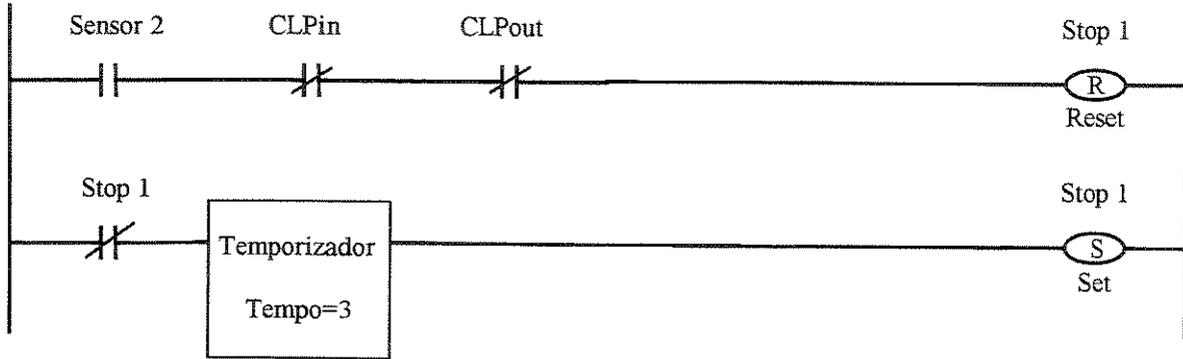


Figura 4.6: Diagrama de Relés do Controle da Trava 1

O terceiro processo que deve ser controlado pelo programa é o funcionamento do elevador. A representação gráfica deste terceiro processo é apresentada na Figura 4.7.

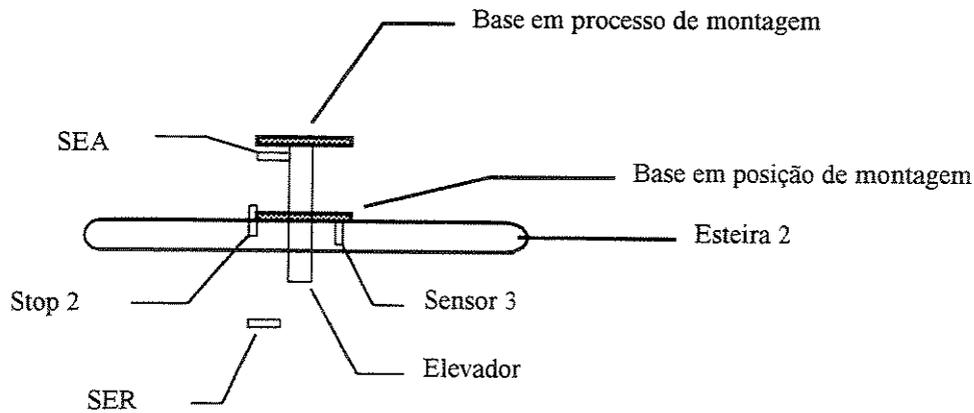


Figura 4.7: Layout para Operação do Elevador

O programa deve levantar o elevador em duas situações:

- toda vez que houver base não montada em posição de montagem;
- ao final do processo de montagem, para poder realizar a descarga da base montada.

O programa deve abaixar o elevador em duas situações:

- após ter fornecido uma base para montagem;

- ao final do processo de montagem, para realizar a descarga da base montada.

O Caderno de Especificações para o terceiro processo (controle do elevador) ficará:

1. Início de processo. Não tem base sendo montada e tem peça em posição de carga. Elevador ficará em 0 até receber sinal do Sensor 3. Ao receber sinal do Sensor 3 e SER, ativar-se o Elevador e CPout;
2. Esperar a CLP pegar a base para abaixar o elevador. Esperar pelo sinais SEA e SGA para desativar o Elevador;
3. Enquanto estiver recebendo os sinais SER e SGF, nada se executa, pois a CLP estará fazendo a montagem da placa. Ao receber os sinais SER e SGF desativar Cpout;
4. A CLP termina o processo de montagem. Ao receber-se o sinal SER e SGA, ativar-se o elevador;
5. Descer o elevador assim que a CLP recolher o braço mecânico (uma espécie de gaveta, que serve para colocar e tirar a base do elevador). Ao receber o sinal SEA e SGF, desativar-se o elevador.
6. a) Quando a peça encostar na esteira desativar a trava 2. Ao receber os sinais Sensor 3 e SER, desativar-se Stop 2.
6. b) Temporizar a desativação do Stop 2 por 3 segundos. Ao ser desativado o Stop 2, esperar 3 segundos para reativá-lo. É o fim do processo.

A partir do Caderno de Especificações, pode-se construir a Tabela de Estados, mostrada a seguir:

Tabela 4.4: Tabela de Estados para Operação do Elevador

Etapa	ENTRADAS						SAÍDAS			AUX
	Sen.3	SEA	SER	SGF	SGA	CPin	Elev.	CPout	Stop2	FM
1	1						1	1		1
2		1			1	0	0			
3			1	1		1		0		
4			1		1	1	1			
5		1		1		1	0			
6a	1								0	0
6b									1	1

Baseado na Tabela de Estados, pode-se construir o programa (em Linguagem de Relés), para Controle do Funcionamento do Elevador, como mostrado na Figura 4.8.

Finalmente, com os três processos prontos, pode-se controlar todo o Transporte de Peças do sistema, que era o objetivo deste item do trabalho.

Utilizando-se a técnica de programação de Diagrama de Relés, pode-se desenvolver um programa que controla os três processos simultaneamente. Apesar do programa funcionar perfeitamente, esta técnica de programação mostrou-se inadequada e ultrapassada. O principal problema encontrado foi a dificuldade de programação, pois existem posições semelhantes na tabela de estados em que o programa deve executar funções distintas. Foi necessário criar variáveis auxiliares, mas assim mesmo, qualquer mudança em um ponto no programa, pode acabar interferindo com outro processo que não tenha nenhuma ligação direta com a mudança em questão.

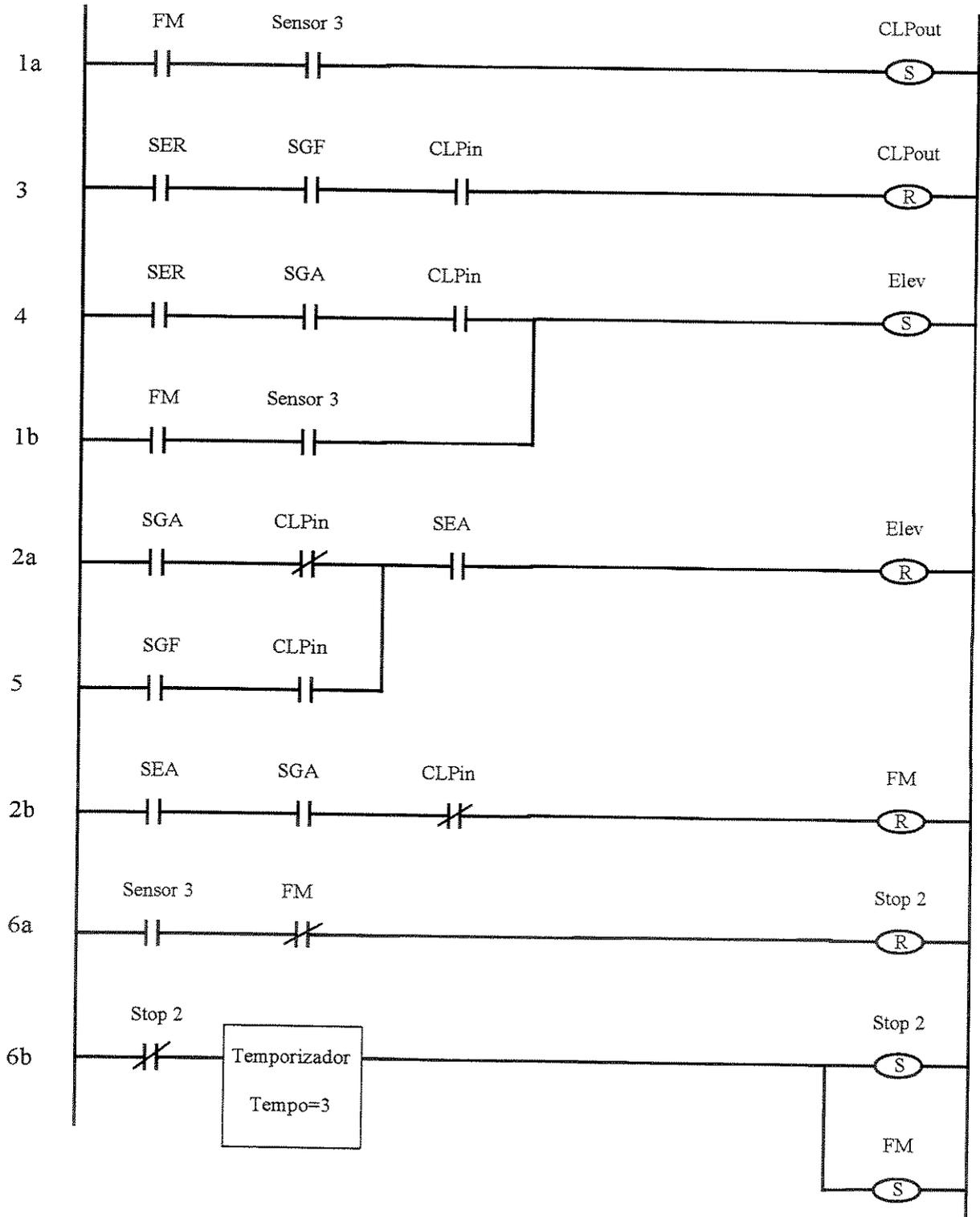


Figura 4.8: Diagrama de Relés do Controle do Elevador

Os diagramas de relés são uma herança dos antigos quadros de relés, utilizados antigamente em processos de automação. Com o desenvolvimento dos Controladores Lógicos Programáveis, os diagramas de relés poderiam ser abandonados, mas os operários já tão acostumados a programação em Diagrama de Relés, preferem continuar trabalhando com este tipo de programação a ter que aprender outro. Dessa forma, os fabricantes de CLPs continuam fabricando equipamentos com este tipo de linguagem de programação.

4.2.2 Programação do transporte de peças em Grafcet

Baseando-se no Caderno de Especificações do primeiro processo (controle do motor da esteira 1), descrito na página 52, pode-se construir um Grafcet para este processo como o descrito na Figura 4.9.

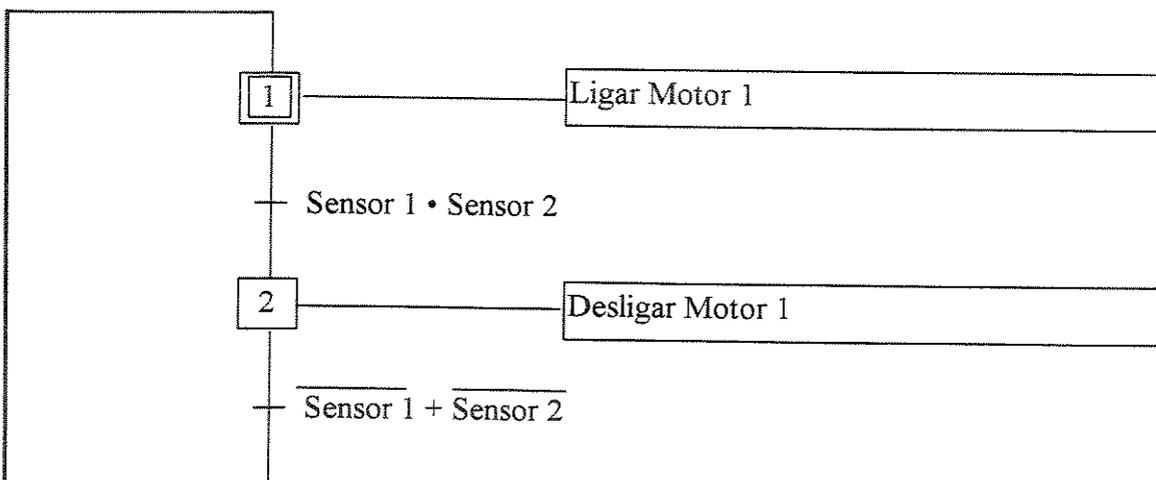


Figura 4.9: Grafcet do Controle do Motor da Esteira 1

Baseando-se no Caderno de Especificações do segundo processo (controle da trava 1), descrito na página 53, pode-se construir um Grafcet para este processo como o descrito na Figura 4.10.

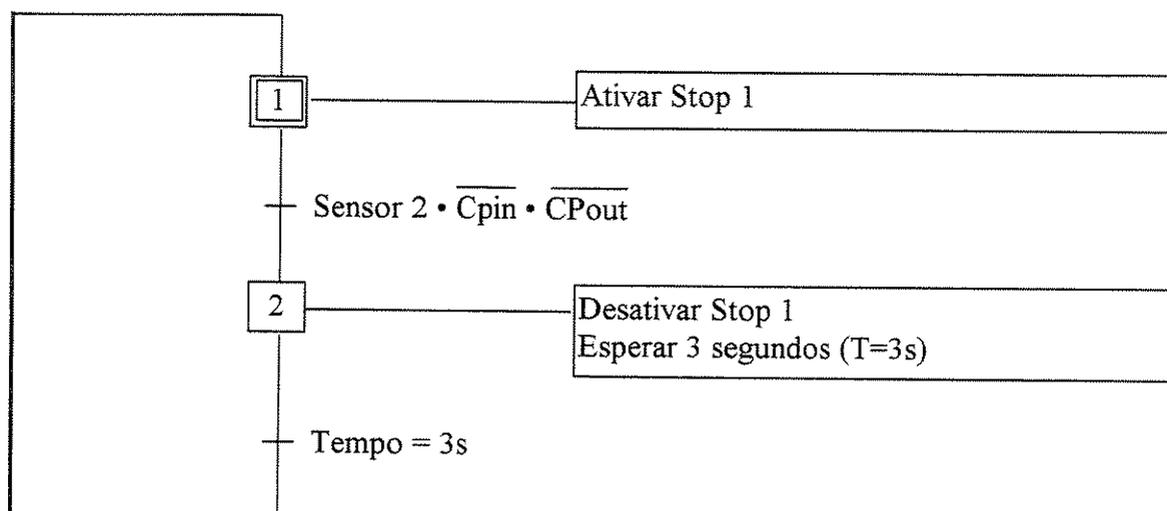


Figura 4.10: Grafcet do Controle da Trava 1

Baseado no Caderno de Especificações do terceiro processo (controle do elevador), descrito na página 56, pode-se construir um Grafcet para este processo como o descrito na Figura 4.11.

Pode-se notar que a programação do processo feita em Grafcet fornece uma descrição quase natural do Caderno de Especificações. A programação feita em Grafcet é muito mais prática e rápida do que a feita em Diagrama de Relés, além de ser mais segura, pois como o programa obedece a um fluxo, não corre-se o risco de ser surpreendido por um estado inesperado dos sensores.

Com este experimento pôde-se desenvolver técnicas de programação de eventos discretos utilizando-se o Visual Basic. Estas técnicas serão aperfeiçoadas no próximo experimento, e serão utilizadas no desenvolvimento do sistema de supervisão para o projeto PIPEFA.

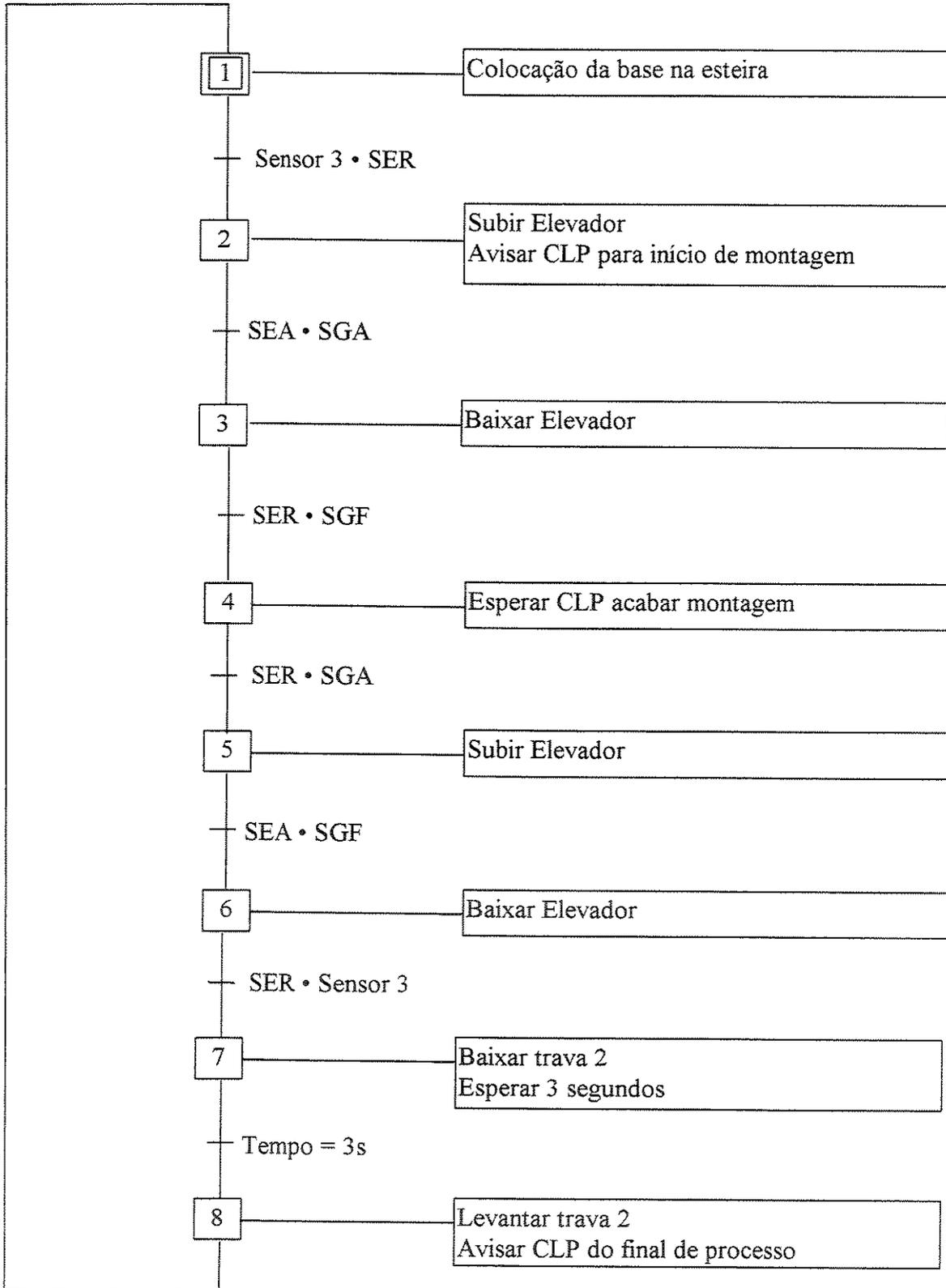


Figura 4.11: Grafcet de Controle do Elevador

4.3 Programa de Controle para Estação de Montagem/Desmontagem

Para aprofundar os estudos sobre a utilização do software Visual Basic no controle de processos, foi proposto um projeto que consiste numa estação de montagem/desmontagem de cubos Legos® com apenas uma esteira transportadora. A esteira recebe uma base de montagem com um código de barras, o qual indicará o tipo de montagem a ser realizada. Poderão ser montados sobre a base de montagem até 3 cubos Legos®. A seqüência de operações do programa é descrita a seguir:

1. Leitura do código de montagem (código de barras);
2. A esteira conduz a base até o elevador;
3. É enviado um sinal para o CLP realizar a montagem;
4. Ao final da montagem o CLP recoloca a base na esteira;
5. A esteira conduz a base montada até seu fim de curso;
6. É invertido o sentido de deslocamento da esteira;
7. A base montada é reconduzida ao elevador;
8. É enviado um sinal ao CLP para iniciar o desmonte;
9. CLP recoloca a base desmontada na esteira;
10. A esteira conduz a base desmontada ao ponto inicial e reinicia-se o processo.

Para controlar este processo usou-se uma técnica de programação de etapas baseada na programação Grafset. Os eventos a serem controlados são descritos em forma de etapas, e só após ser cumprida a condição final de uma etapa, é que o programa passa para a etapa seguinte.

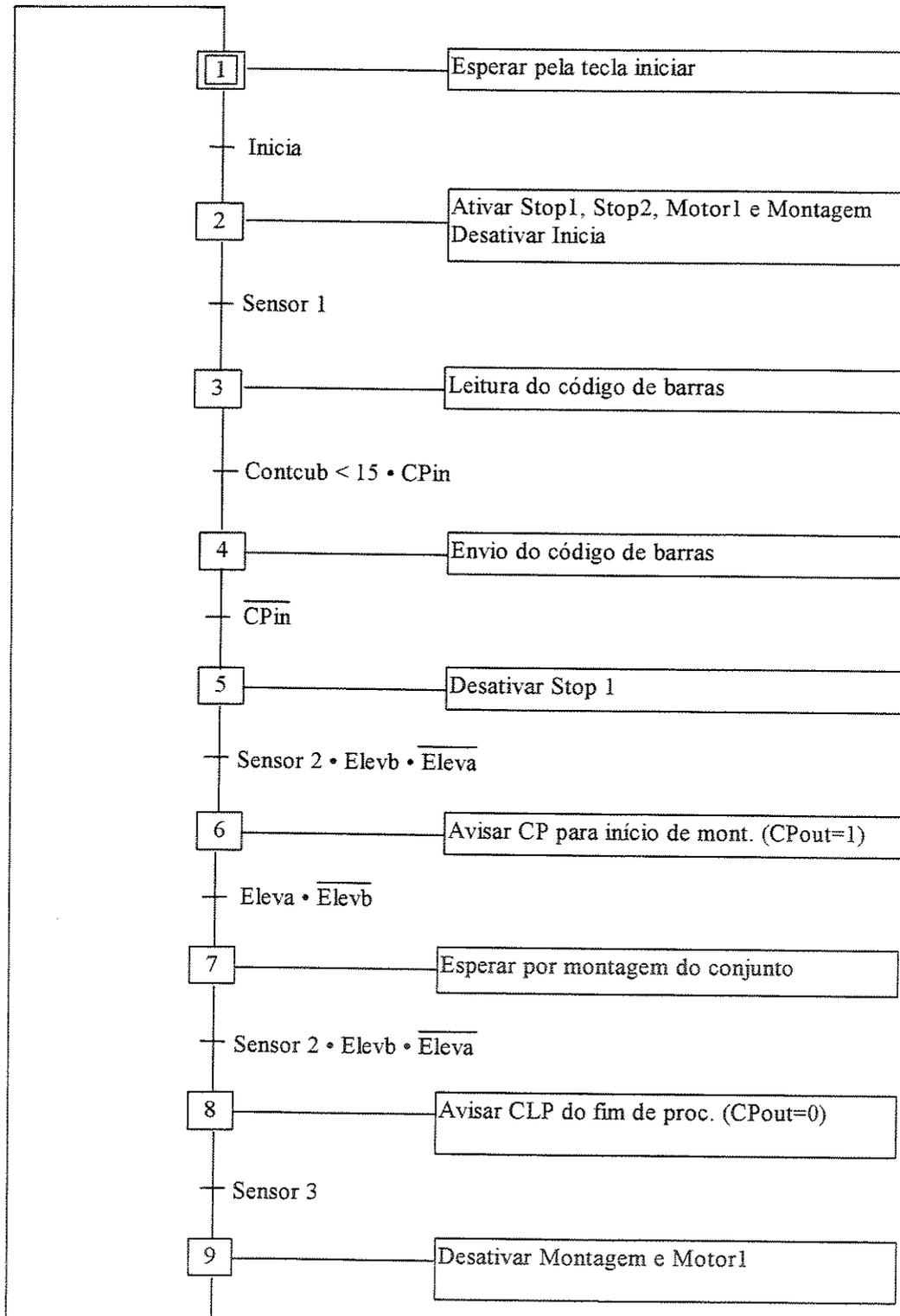


Figura 4.12: Grafcet do Processo de Montagem

Tabela 4.5: Variáveis de Entrada e Saída para a Estação de Montagem/Desmontagem

Variáveis de Entrada		Variáveis de Saída	
Variável	Significado	Variável	Significado
Sensor 1	Sensor Magnético 1	Stop 1	Trava de parada 1
Sensor 2	Sensor Magnético 2	Stop 2	Trava de parada 2
Sensor 3	Sensor Magnético 3	Stop 3	Trava de parada 3
Elevb	Sensor Elev. Recuado	Motor 1	Motor em avanço
Eleva	Sensor Elev. Atuado	Retorno	Motor em retrocesso
CPin	CLP solicita desmont.	CPout	Avisar CLP p/ início de proc.
		Montagem	Processo de Montagem
		Desmont	Processo de Desmonte

Uma vez que a operação de desmonte pode ser realizada por outro posto de trabalho, poderíamos ter a montagem e o desmonte sendo realizadas simultaneamente. Por esta razão fez-se um programa com duas lógicas de controle, capaz de controlar duas operações simultaneamente. A primeira lógica descreve todas as etapas necessárias para a montagem dos cubos na base, e a segunda lógica descreve as etapas necessárias a seu desmonte.

Toda a programação está baseada no “Procedure Timer” do Visual Basic, o qual permite que seja feito um “scanning” contínuo das duas lógicas em tempo real. O primeiro passo da programação do “Procedure Timer” é a leitura do estado das entradas e o armazenamento de seus valores em variáveis, feita através da sub-rotina “Leitura”. O passo seguinte é a utilização da opção “Select Case”, a qual permite que seja realizada apenas a etapa de interesse para ambas as lógicas. Logo, tem-se a utilização de dois controles “Select Case”, o primeiro é o “Select Case Lógica1”, que irá permitir a realização da etapa de interesse para a lógica 1, e o segundo é o “Select Case Lógica2”, que realizará a etapa de interesse para a lógica 2. Logo, cada Etapa do processo de produção é representada por uma instrução Case, e as Transições serão representadas por uma instrução “if”. A última etapa do “Procedure Timer” é o processo de

escrita na placa de entrada/saída, o que permite a atuação do computador no controle do processo, feito pela sub-rotina “Escrita”. A seguir é apresentado um esquema com os principais comandos utilizados no programa:

```

Private Sub Timer1_Timer()
    Leitura                                'Chamada da sub-rotina de ENTRADA DE DADOS
    Select Case Logical1                    'Primeira Lógica: Carregamento
        Case 1                             'ETAPA 1
            ????                            ← 'Coloque aqui as operações desta Etapa
            If Sensor1 Then 'TRANSIÇÃO P/ ETAPA 2
                Logical1 = 2
            End If
        Case 2                             'ETAPA 2
            :
        Case 16                            'ETAPA 16
    End Select
    Select Case Logica2                    'Segunda Lógica: Descarregamento
        Case 1                             'ETAPA 1
            :
        Case 13                            'ETAPA 13
    End Select
    Escrita                                'Chamada da sub-rotina de SAÍDA DE DADOS
End Sub

```

São utilizados os recursos gráficos da plataforma Windows®, presentes no Visual Basic, para apresentar todas as etapas do processo no monitor de vídeo do computador. Tem-se em uma janela a imagem animada com a representação gráfica de todos os sensores, atuadores, deslocamento da base e cubos. Em outra janela tem-se duas tabelas, uma com os estados das entradas e saídas, e outra com a descrição da condição final da etapa que está sendo executada. Esta descrição da condição final da etapa sendo executada é importante para o operador saber o que está acontecendo e qual a Transição necessária para se passar para a etapa seguinte. Por exemplo, para passar a etapa 3 da lógica 1, é necessário que a condição de Transição da etapa 2 - lógica 1 seja cumprida. A condição de Transição da etapa2 - lógica 1 é o sinal do sensor 1 (aviso que existe peça na posição inicial), e enquanto não houver peça na posição inicial tem-se a seguinte mensagem: Lógica 1: Etapa 2 - Esperando por peça no sensor 1. Somente após o

recebimento do sinal do sensor 1 é que o programa passará para a etapa 3 - lógica 1, exibindo assim a seguinte mensagem: Lógica 1: Etapa 3 - Esperando por leitura do código de barras.

Quanto a atuação no processo de montagem/desmontagem, além do controle automático do processo realizado pelo computador, temos ainda a opção de uma tecla pausa, que poderá ser pressionada pelo mouse para parar ou continuar o processo.

A seguir é apresentada uma imagem da tela do computador durante a execução do programa:

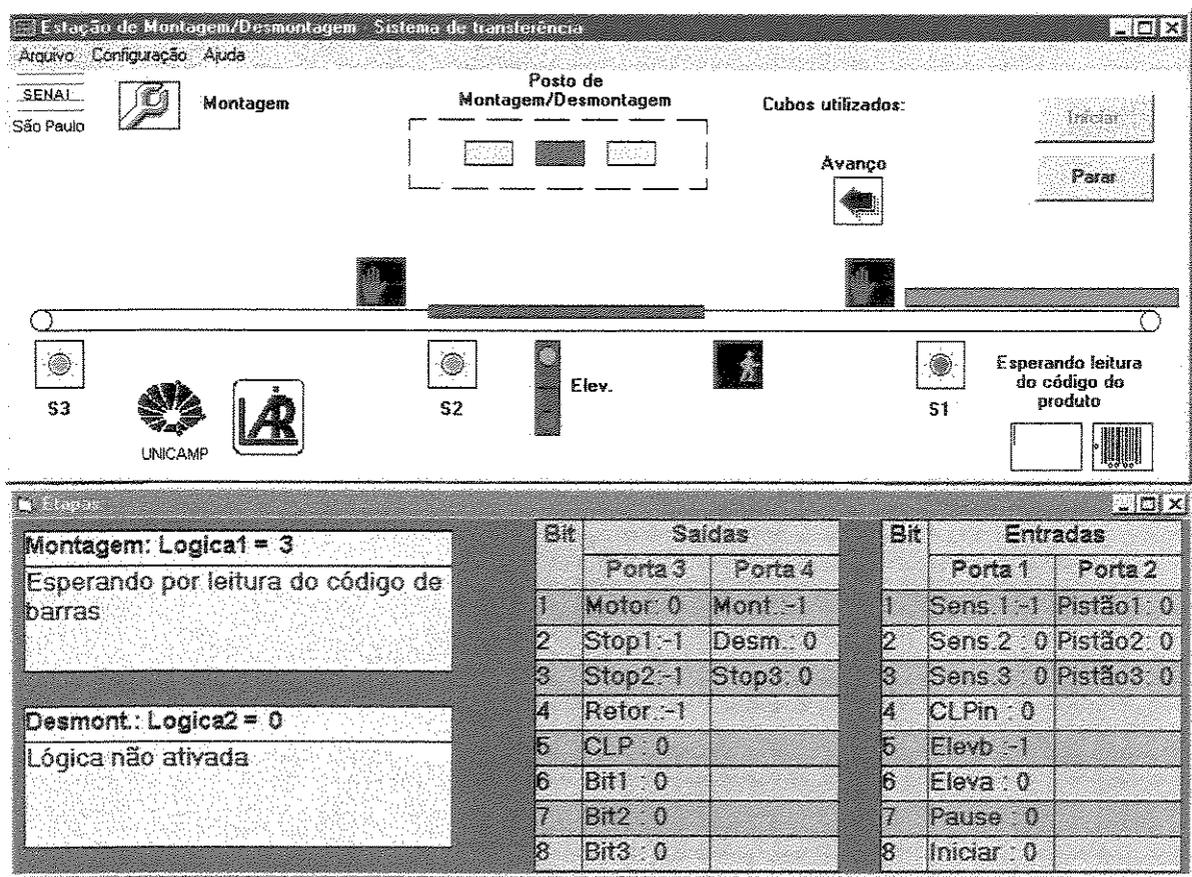


Figura 4.13: Tela de Execução do Projeto Estação de Montagem/Desmontagem

Este projeto foi apresentado na feira de mecânica de 1996, realizada nos dias 20 a 25 de maio, no estande da empresa Robert Shaw. Esta empresa comercializa muito dos dispositivos necessários ao funcionamento deste projeto, como os atuadores pneumáticos, o compressor de ar, o Controlador Programável e alguns sensores. A seguir é apresentada uma foto da Estação de Montagem/Desmontagem em funcionamento durante a feira da Mecânica.

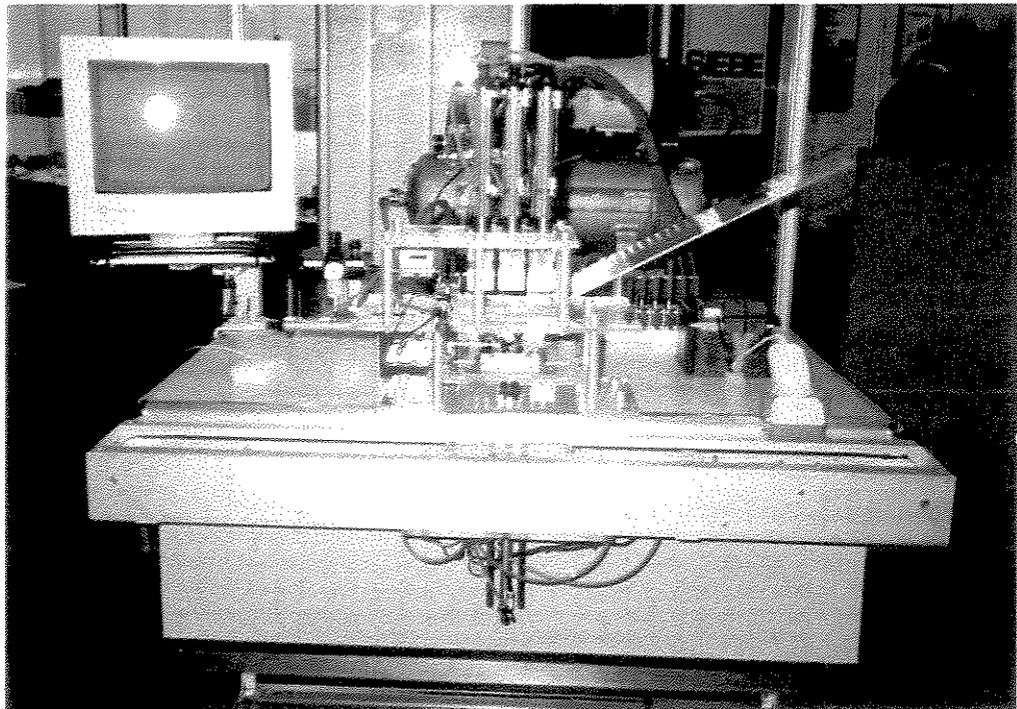


Figura 4.14: Fotografia da Estação de Montagem/Desmontagem

4.4 Implementação de um Programa de Monitoração para Centro de Usinagem

Nesta parte do projeto será contemplado a interconexão entre o nível de Engenharia e o nível de Produção da pirâmide CIM. Será implementado um projeto para o nível de Engenharia da pirâmide CIM ter acesso ao “status” de funcionamento das máquinas CNC, o que representa a comunicação com o sub-nível mais baixo da pirâmide CIM.

Infelizmente ainda não existe um protocolo de comunicação padronizado que permita acesso direto às variáveis de operação das máquinas CNC. Por esta razão, utilizou-se um computador como ponte entre a máquina CNC e o padrão OSI da rede da empresa. Para este computador foi desenvolvido um software aplicativo capaz de monitorar os sinais do CNC e deixando-os disponíveis em rede para acesso remoto. Este acesso remoto foi conseguido graças a utilização dos recursos do sistema DDE (Dynamic Data Exchange), do Windows®.

Para a interface de ligação entre a máquina CNC e o computador foi utilizado um cartão de Entradas/Saídas (I/O) digitais. Tal cartão tem capacidade para 32 bits, divididos em 4 bytes direcionais, ou seja, cada byte (de 8 bits) pode ser definido para uma das duas operações: Entrada ou Saída (I/O). Para ter acesso aos dados adquiridos pelo cartão I/O, foi utilizada uma biblioteca de vínculo dinâmico (DLL – Dynamic Link Library), obtida através dos trabalhos de Bulback (1998). Esta biblioteca permite ao Visual Basic ter acesso aos dados das portas de entrada/saída do microcomputador através de seu endereçamento. Pode-se também utilizar a porta paralela para controle e aquisição de dados. Ela proporciona oito saídas TTL, cinco entradas e quatro sinais bidirecionais, facilmente utilizáveis através da utilização correta de seus endereços. Anderson (1996) relata vários modelos de desenvolvimento de programas para utilização da porta paralela. Ele mostra que a porta paralela pode também ser usada para controle de motores DC, motores de passo, controle remoto por rádio e infravermelho, multiplexador e conversor D/A e A/D. No Anexo V é mostrado como a porta paralela deve ser configurada para aquisição e envio de dados, e no Anexo VI é exemplificado como deve ser programada a comunicação de dados. O programa exemplificado no Anexo VI utiliza a biblioteca de vínculo dinâmico desenvolvida por Bulback (1998), mas outra biblioteca muito parecida com esta pode ser encontrada nos trabalhos de Hart (1999).

Para a ligação do computador com outras áreas da empresa foi utilizada uma placa de rede padrão Ethernet NE2000, com capacidade de 10 Mbits/s. O padrão de rede utilizado foi o ponto a ponto (“peer to peer”) e o protocolo de comunicação foi o NetBEUI para Clientes da Redes Microsoft. O sistema operacional utilizado foi o Windows for Workgroups®.

Os dados que devem estar disponíveis na pirâmide CIM, são as entradas e saídas do CLP da máquina CNC. As entradas são sinais (dos sensores ou outros dispositivos), que são enviados ao CLP do CNC. As saídas são sinais enviados do CLP do CNC para os atuadores. As operações que poderão ser monitoradas pelo programa de supervisão são:

Sinais das entradas e seus endereços que podem ser monitorados:

- Status da máquina:
 - Nível de lubrificação (X0.0);
 - Sensor de lubrificação (X0.1);
 - Sinal de sobrecarga (X0.2);
 - Sensor porta fechada (X2.2);
 - Sensor de presença de carga (X2.4);
 - Sensor de presença de descarga (X2.5);
 - Sensor de referência X (16.5);
 - Sensor de referência Y(17.5);
 - Sensor de referência Z (18.5);

- Sinal de emergência (21.4).

- Ferramental:
 - Sensor de Ferramenta Travada (X0.3);
 - Sensor de Ferramenta Destravada (X0.4);
 - Contator liga Motor trava/destrava ferramenta (X0.5);
 - Sensor carro do T.A.F. avançado (X0.7);
 - Sensor carro do T.A.F. recuado (X2.0);
 - Contator liga Motor da pré-seleção ferramenta (X2.1);
 - Sensor da porta pré-seleção (X2.3);
 - Sensor de referência do TAF (Troca Automática de Ferramentas);

- Troca de Pallet:
 - Sensor do pistão de carga avançado (X2.6);
 - Sensor do pistão de carga recuado (X2.7);
 - Sensor do pistão de levantamento avançado (X4.6);
 - Sensor do pistão de levantamento recuado (X4.0);
 - Sensor do pistão de descarga avançado (X4.1);

- Sensor do pistão de descarga recuado (X4.2);
- Sensor do pistão de transferência avançado (X4.3);
- Sensor do pistão de transferência recuado (X4.5);
- Sensor da porta do pallet aberta (X16.1);
- Sensor da porta do pallet fechada (X16.2).

Sinais das saídas e seus endereços que podem ser monitorados:

- Status da máquina:
 - CNC enable (Y48.0);
 - Alarme sonoro (Y49.3);
 - Lâmpada indicadora de trabalho (Y50.2);
 - Lâmpada indicadora de atenção (Y50.3);
 - Lâmpada indicadora de parada (Y50.5);
 - Lâmpada permite partida (Y80.2);
 - Eletroválvula abre porta automática (Y52.5);
 - Eletroválvula fecha porta automática (Y52.6).
- Troca de ferramentas:

- Eletroválvula destrava ferramenta (Y48.6);
- Eletroválvula limpeza de ferramentas (Y48.7);
- Eletroválvula avança carro do T.A.F. (Y49.0);
- Eletroválvula recua carro do T.A.F. (Y49.1);
- Refrigeração/Lubrificação:
 - Contator do Motor de refrigeração (Y48.1);
 - Contator do Motor de lubrificação (Y48.2);
 - Eletroválvula refrigeração por ar de corte (Y49.7);
 - Eletroválvula refrigeração por chuveiro (Y50.0);
 - Eletroválvula refrigeração pelo centro da ferramenta (Y52.1);
- Troca de Pallet:
 - Eletroválvula pistão de carga avança (Y52.7);
 - Eletroválvula pistão de carga recua (Y80.7);
 - Eletroválvula pistão de descarga avança (Y53.0);
 - Eletroválvula pistão de descarga recua (Y82.0);
 - Eletroválvula pistão abre porta do pallet (Y53.1);

- Eletroválvula pistão fecha porta do pallet (Y53.2);
- Eletroválvula pistão levantamento avança (Y53.3);
- Eletroválvula pistão levantamento recua (Y53.4);
- Eletroválvula pistão de transferência avança (Y53.5);
- Eletroválvula pistão de transferência recua (Y82.1);
- Eletroválvula de limpeza (Y53.6);
- Eletroválvula trava a mesa (Y53.7);
- Eletroválvula destrava a mesa (Y80.0);
- Eletroválvula trava permanente a mesa (Y80.1).

Para validação do experimento, foi desenvolvido um programa em Visual Basic, para monitoração do sistema de troca de pallet do Centro de Usinagem Polaris V400. Este programa monitora todas as entradas e as saídas envolvidas do processo de troca de pallet (descrito acima), representando graficamente seus estados na tela do computador. São representados na tela do computador os seguintes sinais dos sensores (entradas): pistões de carga, levantamento, descarga, transferência e porta do pallet nos estados avançado e recuado. São também representados os sinais dos seguintes atuadores (saídas): eletroválvulas nos estados avança e recua os pistões de carga, levantamento, descarga, transferência e porta do pallet, eletroválvula de trava, destrava e trava permanente da mesa, além da eletroválvula de limpeza.

Com este sistema pode-se acompanhar pelo computador, em tempo real, todas as etapas do processo de troca de pallet, como mostrado na Figura 4.15, a seguir:

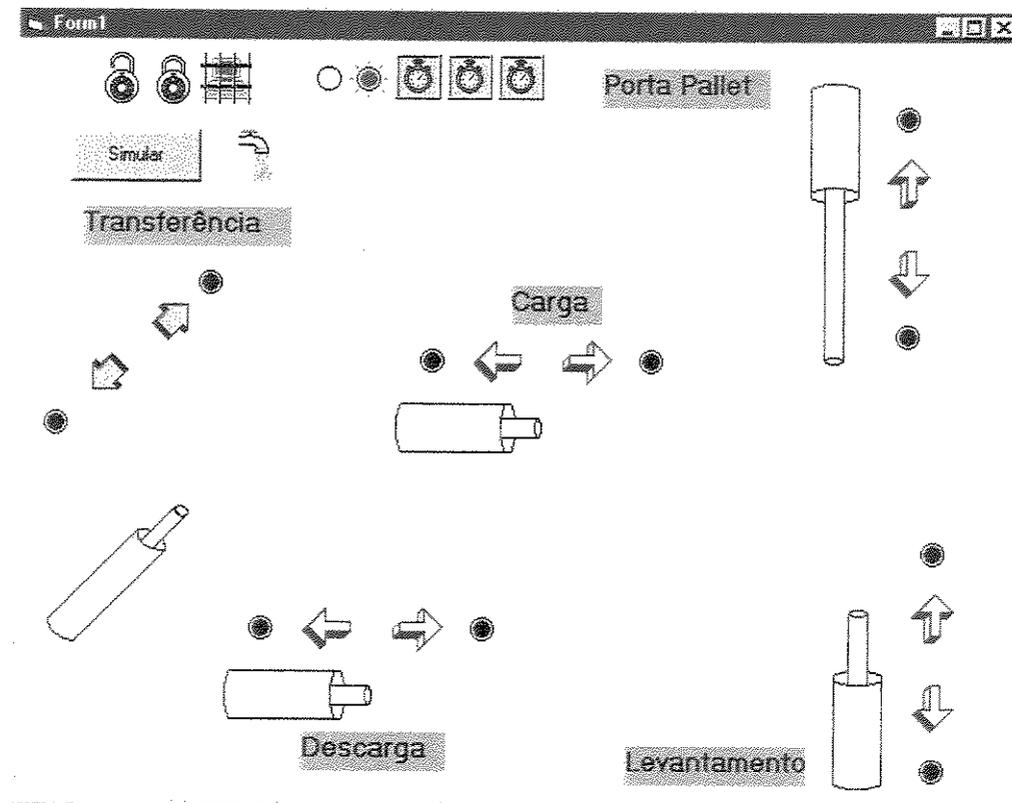


Figura 4.15: Monitoração do Processo de Troca de Pallets

Este software aplicativo foi desenvolvido com animação gráfica, de forma que, além dos sinais dos sensores e atuadores, que são mostrados em tempo real, também é apresentada uma animação gráfica do movimento dos cilindros, que não é feita em tempo real. Os sinais dos sensores são apresentados na tela em forma de LED verde para o nível lógico 0 e LED vermelho para o nível lógico 1. As eletroválvulas dos pistões são apresentadas em forma de seta (indicando a direção do movimento); para este caso o sistema lê o sinal de saída do CLP para os atuadores, e é mostrado na tela apenas as setas que representam sinal alto (nível lógico 1). As demais eletroválvulas são apresentadas com ícones característicos, como por exemplo um cadeado aberto para mesa destravada, cadeado fechado para mesa travada, grades para mesa travada permanentemente e torneira aberta para eletroválvula de limpeza ligada, como pode ser observado na Figura 4.15. O programa também possui uma tecla para simulação do sistema; neste caso os sinais dos sensores e atuadores são forçados para que apareçam na tela como se

estivesse ocorrendo uma troca de pallets. Para funcionar desta forma o programa foi escrito com três temporizadores. O primeiro temporizador serve para fazer a aquisição dos dados e apresentá-los na tela em tempo real. O segundo temporizador serve para realizar a animação gráfica do deslocamento dos pistões. O terceiro temporizador serve para a simulação do processo.

Este tipo de utilização de sistemas abertos, para controle de célula flexível de manufatura, é amplamente comentado por Becerra (1993).

Neste capítulo foram apresentadas três aplicações a sistemas automatizados de produção. Foram utilizadas diferentes ferramentas para descrição de sistemas sequenciais, e verificada suas vantagens e limitações. Foram desenvolvidas técnicas de programação de eventos discretos utilizando-se o Visual Basic. Neste capítulo foram utilizadas animações gráficas e desenvolvidas interfaces homem-máquina, necessárias ao desenvolvimento do simulador educativo visto no capítulo anterior.

No próximo capítulo serão vistos aspectos na utilização de banco de dados SQL, utilizados para melhorar a conectividade e a flexibilidade do sistema. Será desenvolvido e implementado um sistema de supervisão para o projeto PIPEFA. Será demonstrada a grande importância dos protocolos de comunicação na união dos níveis da pirâmide CIM, justificando assim, sua necessidade de padronização.

Capítulo 5

Implementação do Sistema de Supervisão ao Projeto PIPEFA

Neste capítulo serão utilizadas as ferramentas e técnicas de programação de eventos discretos, estudadas no capítulo anterior, para implementação do sistema de supervisão para o projeto PIPEFA. Este capítulo explicitará a importância na utilização de sistemas abertos para desenvolvimento de aplicações, deixando como forma inadequada de implementação, aquelas específicas e rígidas. Será vista a importância dos protocolos de comunicação, na integração dos elementos constituintes do sistema automatizado de produção.

Para supervisionar o funcionamento da plataforma foi desenvolvido um aplicativo, utilizando-se uma linguagem de programação “popular”, de baixo custo e amplamente difundida no mercado, que é o Visual Basic.

A programação do sistema de supervisão foi desenvolvida de uma forma multitarefa, utilizando-se o comando DoEvents dentro dos Loops do programa para se evitar perda de dados durante a atualização de tela, e monitorando-se cada estágio do processo independentemente. Com isto, pode-se ter várias peças sendo montadas ao mesmo tempo, não sendo permitida a presença de duas peças no mesmo estágio simultaneamente.

A idéia principal deste sistema de supervisão é permitir o acompanhamento do processo de produção a partir de estações remotas. Inicialmente, tentou-se utilizar recursos do DDE para LAN (Local Area Network), mas isto limitava os aplicativos que poderiam consultar os dados, pois deveriam ser aplicativos do Windows® que suportassem DDE. A segunda tentativa foi utilizar a área de armazenamento temporário do Windows®. Isto retirou a limitação dos

aplicativos de suportarem DDE. Outra tentativa foi utilizar um arquivo ASCII para armazenar os dados importantes. Finalmente, decidiu-se utilizar um banco de dados para armazenamento das informações sobre o processo de montagem. Optou-se por um banco de dados padrão Access, que trabalha de forma integrada ao Visual Basic e pode ser compartilhado em um servidor Web. Dessa forma, conseguiu-se desenvolver, de uma forma inovadora, um sistema que permita a observação e registro do sistema, não só a partir da rede local, mas de qualquer lugar que tenha acesso à Internet, por qualquer aplicativo que trabalhe com Access.

Fazendo uma comparação com a hierarquia da pirâmide CIM, o sistema de produção ficou estruturado de maneira análoga, com o Sistema de Gestão de Produção encabeçando a pirâmide e o nível de produção em sua base, como mostrado na Figura 5.1. Quando é emitida uma nova ordem de produção pelo Sistema de Gestão de Produção, esta é acrescentada ao banco de dados. Uma vez que o sistema de supervisão é informado que há um pedido de produção pendente, libera sua execução, na medida do possível.

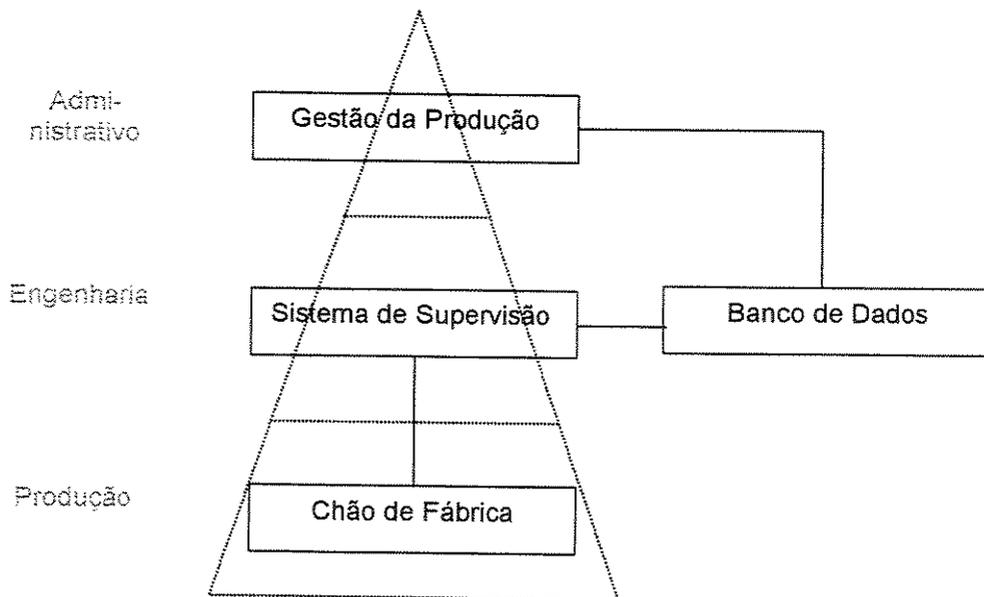
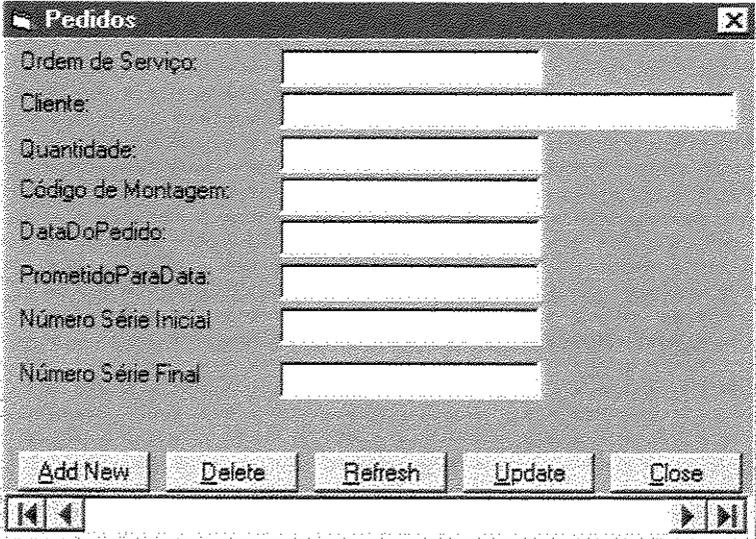


Figura 5.1: Estruturação do Sistema de Supervisão

Por outro lado, o nível de produção envia informações, ao sistema de supervisão, a respeito do andamento dos processos. Tais informações são acrescentadas ao banco de dados em tempo real.

Dessa forma, o banco de dados reflete, não só o estado do andamento dos processos, como também os produtos que ainda serão montados.

Além das ordens de serviço enviadas pela Gestão da Produção (Figura 5.2), o sistema de supervisão recebe informações sobre o funcionamento do sistema (variáveis do sistema) e da interface homem/máquina. A comunicação com a Gestão da Produção é feita pelo banco de dados; a comunicação com o chão de fábrica é feita pela interligação com o CLP mestre e a comunicação com o operador é feita pela interface gráfica do computador. Diante de todas essas informações o sistema de supervisão consegue emitir as ordens de fabricação para cada posto de trabalho, como mostrado na Figura 5.3:



The image shows a screenshot of a software window titled "Pedidos" (Orders). The window contains several input fields for data entry:

- Ordem de Serviço:
- Ciente:
- Quantidade:
- Código de Montagem:
- DataDoPedido:
- PrometidoParaData:
- Número Série Inicial
- Número Série Final

At the bottom of the window, there are five buttons: "Add New", "Delete", "Refresh", "Update", and "Close". Below the buttons is a navigation bar with left and right arrow icons.

Figura 5.2: Formulário para Emissão de Ordens de Serviço

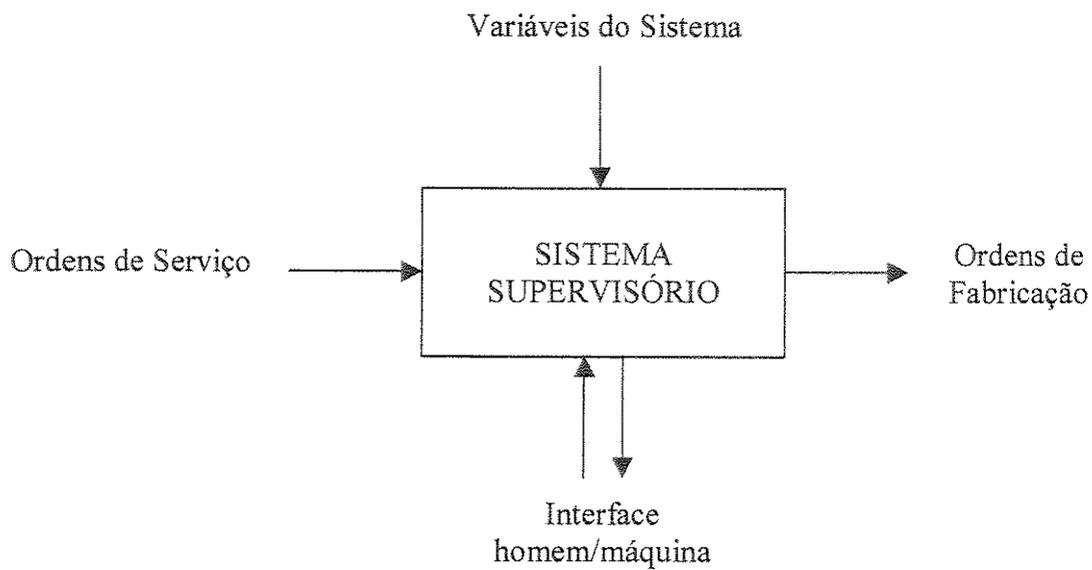


Figura 5.3: Entradas/Saídas do Sistema de Supervisão

Para realizar as operações de montagem, o funcionamento do sistema de supervisão foi dividido em dez estágios, como mostrado Figura 5.4:

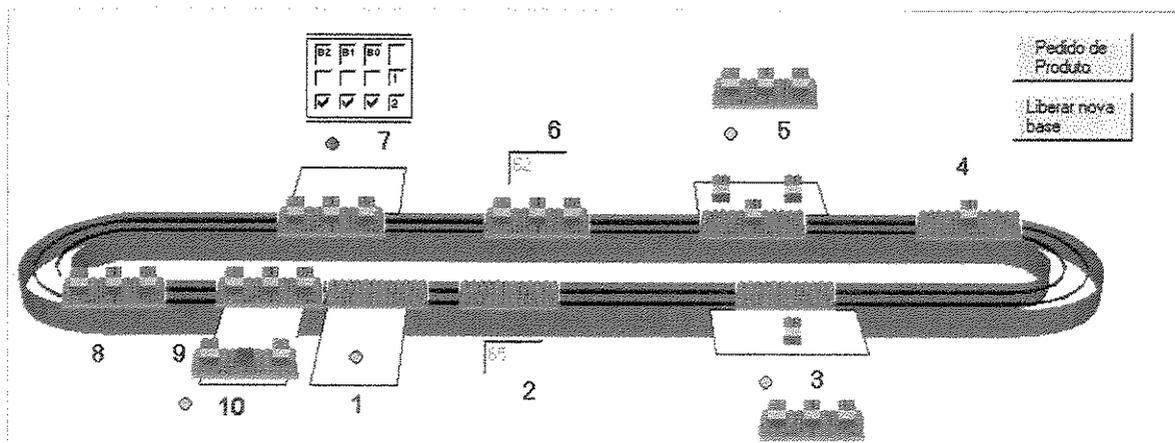


Figura 5.4: Etapas de Produção do Projeto PIPEFA

1. No primeiro estágio, tem-se o posto de carregamento, onde as bases são automaticamente colocadas na esteira transportadora.
2. O segundo estágio serve para leitura do código de barras.
3. No terceiro estágio, tem-se o posto de Montagem/Desmontagem Central, responsável pela montagem/desmontagem dos cubos centrais.
4. O quarto estágio organiza o posicionamento das bases na esteira transportadora.
5. No quinto estágio, tem-se o posto de Montagem/Desmontagem Lateral, responsável pela montagem/desmontagem dos cubos laterais.
6. No sexto estágio realiza-se a segunda leitura do código de barras.
7. No sétimo estágio, tem-se o posto de inspeção, responsável pela verificação da montagem.
8. O oitavo estágio, novamente organiza o posicionamento das bases na esteira transportadora.
9. No nono estágio, tem-se o posto de descarregamento, responsável pela retirada das bases da esteira transportadora.
10. O décimo estágio sinaliza o final de processo, a peça está pronta e armazenada.

As informações importantes a respeito do processo de montagem, que constam do banco de dados são:

1. Número seqüencial do produto (número de série);
2. Hora, minuto e segundo de início de cada estágio;

3. Código de montagem;
4. Código de barras;
5. Código de inspeção;
6. Informação sobre o tipo de operação: montagem ou desmontagem.

Um exemplo do banco de dados é mostrado na Tabela 5.1. Neste exemplo constata-se que o produto com número de série 9 foi montado em 1,4 minutos; o produto com número de série 10 está em fase de montagem no estágio 5; o produto com número de série 11 está em fase de montagem no estágio 3.

Tabela 5.1: Banco de Dados Utilizado pelo Sistema de Supervisão

nséri	Est1	Est2	Est3	Est4	Est5	Est6	Est7	Est8	Est9	Est10	CMont	CBar1	CBar2	CInsp	Montar
9	8:59:43	8:59:57	9:00:15	9:00:24	9:00:27	9:00:34	9:00:56	9:01:00	9:01:04	9:01:07	25	55	55	25	SIM
10	9:02:35	9:03:35	9:04:17	9:04:33	9:04:36	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	24	56	0	0	SIM
11	9:04:52	9:05:03	9:05:14	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	19	57	0	0	SIM

O número de série (campo nséri), ou número seqüencial do produto, indica quantos produtos foram fabricados. Cada vez que é emitido um pedido de fabricação de um produto, ele recebe um número seqüencial.

O momento de início de cada estágio (campo Est_) é importante para o setor de planejamento do processo fazer o controle de tempos de produção, otimização do processo, estimativa dos tempos de produção e verificação do andamento da produção. Quando um novo pedido de fabricação é emitido, estes campos são zerados.

O código de montagem (Campo C.Mont) indica o tipo de produto a ser produzido. Podem ser produzidos 27 tipos de produtos diferentes, de 0 a 2 cubos na lateral direita, esquerda e

posição central ($3 \times 3 \times 3 = 27$ tipos de produtos), como mostrado no Anexo I. O código de montagem pode ser representado por um número base dez ou base três. A utilização de um número base três facilita a visualização, pois utiliza-se o primeiro dígito para definir a lateral esquerda, o segundo dígito para definir a posição central e o terceiro dígito para a lateral direita. Por exemplo, se houver um cubo na lateral esquerda, nenhum no centro e dois na lateral direita, tem-se em representação base três: 102_3 . A conversão para decimal é feita da seguinte forma:

$$X = \underline{1} \times 3^2 + \underline{0} \times 3^1 + \underline{2} \times 3^0$$

$$X = 11_{10}$$

A conversão de decimal para base 3 pode ser feita como mostrado na Figura 5.5.

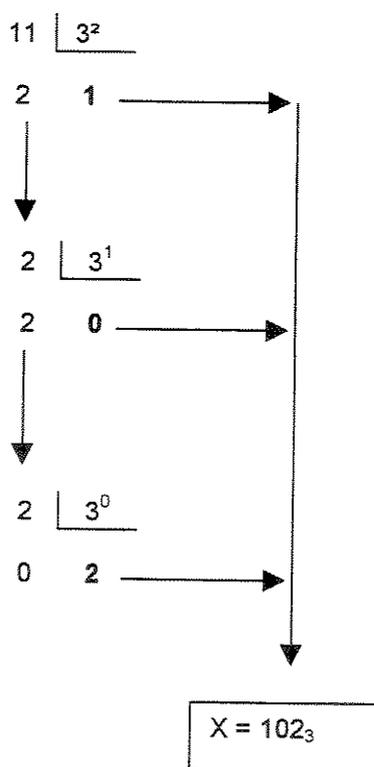


Figura 5.5: Conversão Base 10 para Base 3

Este código de montagem será associado com o código de barras presente na placa base. Na primeira vez que o sistema lê o código de barras (campo C.Bar1), ele é associado ao código

de montagem. Na próxima vez que ele é lido (campo C.Bar2), já se sabe qual produto que está sendo montado sobre aquela placa base.

A próxima informação presente no banco de dados é o código de inspeção (campo C.Insp), que representa a produto que foi montado. Ele é comparado com o código de montagem para verificação do produto.

Finalmente, a última informação do banco de dados é o tipo de operação (campo Montar) a ser realizada, indicando se é a montagem ou o desmonte do produto que deve ser feito.

Vejamos um exemplo de funcionamento do sistema de supervisão:

1. O Sistema de Gestão de Produção emite um pedido de fabricação de um determinado produto. Automaticamente a especificação do produto a ser fabricado, é acrescentada ao campo CMont do banco de dados. Neste instante, também automaticamente é gerado um número de série para este produto, o qual é acrescentado ao campo nséri do banco de dados. Os campos Est_ do banco de dados são zerados;
2. Recebendo um novo pedido de fabricação e caso não haja produto nos estágios de montagem 1 e 2, o sistema de supervisão autoriza a fabricação do novo produto. Neste instante é acrescentada a informação da hora:minuto:segundo no campo Est1 do banco de dados. É enviado um sinal ao posto de carregamento autorizando a liberação de nova base de montagem;
3. É lido o código de barras da base que foi liberada para montagem. Neste instante é acrescentada a informação da hora (hh:mm:ss) no campo Est2, e armazenado o código de barras no campo CBar1;
4. A base passa pelo sensor do posto de montagem central. É acrescentada a informação da hora no campo Est3. Se houver necessidade de montagem central ela será realizada;
5. A base passa pelo sensor da curva 1. É acrescentada a informação da hora no campo Est4;

6. A base chega ao posto de montagem lateral. É acrescentada a informação da hora no campo Est5. Se houver necessidade de montagem de cubos ela será realizada;
7. É lido e armazenado o valor do segundo código de barras no campo CBar2. Também é acrescentada a informação da hora no campo Est6;
8. A base chega ao posto de inspeção. É acrescentada a informação da hora no campo Est7. É lido o código de inspeção e armazenado seu valor no campo CInsp. Dependendo do resultado do controle de qualidade, o produto pode ser classificado como bom, seguindo normalmente para o posto de armazenamento; considerado reaproveitável, sendo acrescentada a informação de desmontagem no campo Montar; ou considerado refugo, sendo descartado e jogado fora;
9. Caso o produto siga para o estágio seguinte, chegará ao sensor da curva 2, sendo registrada a informação da hora no campo Est8;
10. O produto chega ao posto de descarregamento. É acrescentada a informação da hora no campo Est9. Caso o campo Montar estiver com a informação sim, é enviado sinal para o posto de Descarregamento efetuar o armazenamento. Caso contrário, o produto seguirá para desmontagem;
11. É recebido sinal do posto de armazenamento (a ser acrescentado ao projeto) significando produto pronto e armazenado. É acrescentada a informação da hora no campo Est10.

O sistema de supervisão também é responsável pela geração de alarmes. Existem dois conjuntos de alarmes gerados pelo sistema de supervisão. O primeiro conjunto de alarmes são os sinais enviados de cada posto, significando posto em pane. Normalmente os sistemas de supervisão trabalham com uma tela só para alarmes, mas isto prejudica a interface homem-máquina. Optou-se então por colocar um “Led” junto ao desenho de cada posto, que oscila verde/vermelho toda vez que o posto estiver em pane. Além disto é emitido um aviso sonoro para alertar o operador. O segundo conjunto de alarmes são os sinais que detectam desvio na operação dos postos. Para estes problemas são geradas mensagens e soluções específicas. Por

exemplo, o código de barras 2 difere do código de barras 1; neste caso é gerada uma mensagem do tipo: “Atenção: Cód. Bar. II difere do Cód. Bar. I” . Outro exemplo particularmente interessante é quando uma base chega a um posto; o sistema primeiro verifica se realmente existem bases a serem montadas, e caso não existam exibe a seguinte mensagem: “Erro de Montagem: Não era esperada base no Est_”; depois o sistema verifica se foi finalizada montagem no estágio anterior, e caso contrário exibe a seguinte mensagem: “Erro de Montagem: Etapa anterior, Est_, não realizada. Base em posição indevida”.

O sistema de supervisão foi desenvolvido utilizando-se um sistema de “scanning” contínuo, ou seja, todas as etapas são verificadas dentro de um ciclo de “scanning”, ficando assim independentes. Na programação do sistema de supervisão, não existem comandos de interrupção de processamento, de forma que o sistema está sempre recebendo e processando as informações provenientes dos postos de trabalho. Este “scanning” contínuo compreende as seguintes etapas:

1. Leitura e tradução das entradas;
2. Geração de alarmes;
3. Execução simultânea de vários processos;
 - 3.1. Carregamento;
 - 3.2. Leitura do código de barras I;
 - 3.3. Mont./Desmont. Central;
 - 3.4. Passagem pela Curva 1;
 - 3.5. Mont./Desmont. Lateral;
 - 3.6. Leitura do código de barras II;

- 3.7. Leitura do código de inspeção;
 - 3.8. Passagem pela Curva 2;
 - 3.9. Descarregamento;
 - 3.10. Armazenamento;
4. Tradução das saídas e envio dos dados.

A Figura 5.6, representa o “scanning” contínuo realizado pelo programa. Nota-se que os dez estágios que foi dividido o projeto PIPEFA são executados simultaneamente. Isto significa que poderíamos ter até 10 produtos sendo montados ao mesmo tempo.

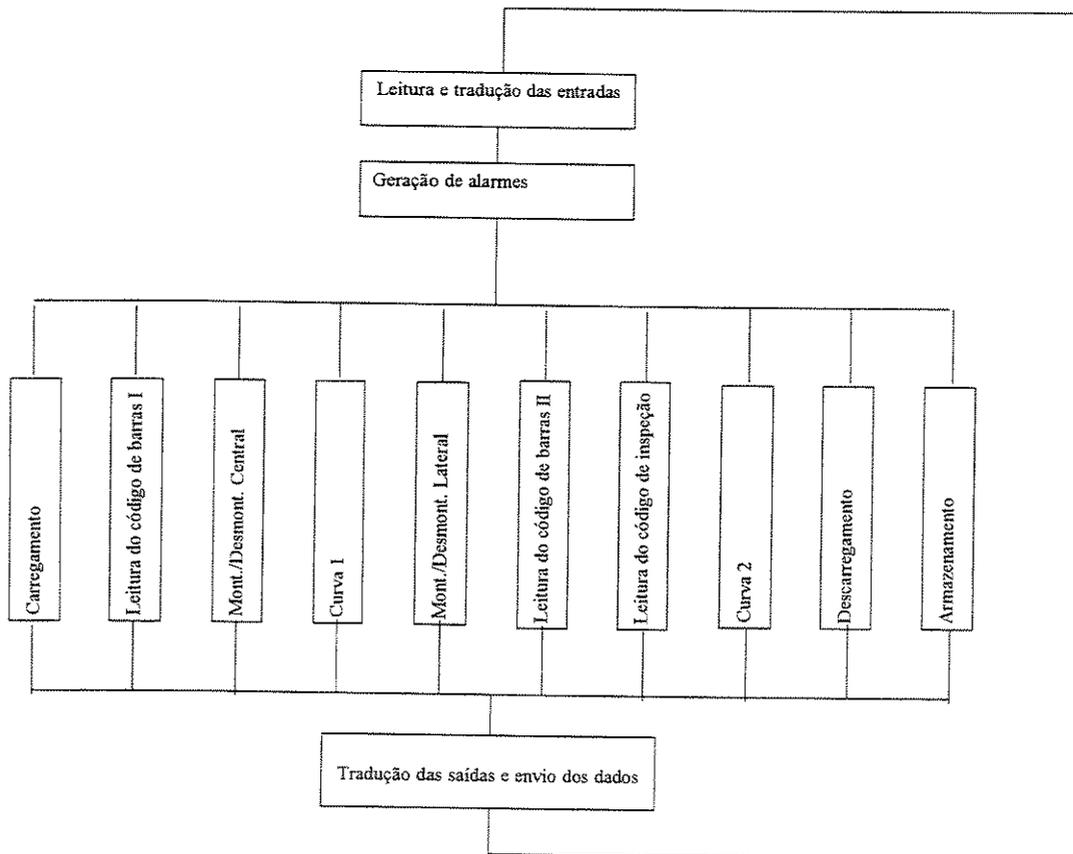


Figura 5.6: Processos Simultâneos Realizados no PIPEFA

Cada um dos processos realizados simultaneamente é representado por um grafcet, e todos esses grafcet são executados simultaneamente. O grafcet completo, com os dez processos de montagem é apresentado no Anexo III. O tempo médio (medido durante a realização de testes) para a realização de cada “scanning” é de 112 milissegundos, ou seja, a cada décimo de segundo aproximadamente, o programa verifica todas as etapas e toma as ações que forem necessárias.

O termo tradução das entradas e saídas se refere a utilização de mnemônicos. A utilização somente de mnemônicos em todo o programa, dificultaria o processo de aquisição e envio de dados, por isso são utilizados vetores binários de entradas (Bin()) e saídas (Bout()) no processo de aquisição de dados; mas a utilização destes vetores ao longo do programa, dificultaria a sua criação e verificação, uma vez que necessita-se lembrar a qual sinal se refere cada um dos valores. Por exemplo, seria necessário lembrar que Bin(1) refere-se ao sensor do primeiro código de barras. Por esta razão utilizou-se os vetores binários para fazer a aquisição e envio de dados e códigos mnemônicos no resto do programa. Chamou-se de tradução ao processo de associação dos vetores binários aos códigos mnemônicos, como mostrado abaixo:

'Tradução das Entradas

SENS_COD1 = Bin(1)	'Sensor Cod. Bar. 1
SENS_CENT = Bin(2)	'Sensor Posto Central
SENS_CURVA1 = Bin(3)	'Sensor da Curva1
Alarm_Car = Bin(9)	'Alarme do P. Carregamento
Alarm_Cent = Bin(10)	'Alarme do P. Mont. Central
B11 = Bin(14)	'Código de Inspeção
B12 = Bin(15)	'Código de Inspeção

'Tradução das Saídas

Bout(1) = CARREGA	'Ordem de inicio de processo
Bout(2) = LATERAL_D	'Aviso que haverá M/D da Lateral Direita
Bout(3) = LATERAL_E	'Aviso que haverá M/D da Lateral Esquerda
Bout(4) = CENT	'Aviso que haverá M/D Central
Bout(8) = CENT_MD	'Aviso se haverá montag. Ou desmont.

Bout(9) = CENT_12 'Aviso se será montado 1 ou 2 cubos

A descrição completa de todos os mnemônicos utilizados, seu significado, a especificação da borneira e do borne de referência são apresentados no Anexo II.

Para verificação do sistema de aquisição de dados, o programa dispõe de uma tela com o estado e o código mnemônico de todas as variáveis utilizadas no processo, como mostrado na Figura 5.7. Esta tela é utilizada para testes do programa, e por ela pode-se verificar possíveis falhas no programa principal. Foi criada uma outra tela apenas com os sinais das saídas, pois durante a utilização normal do programa, a tela anterior atrapalha a visualização. Esta nova tela, mostrada na Figura 5.8, apenas com os valores das saídas, é exibida no canto do vídeo, não atrapalhando a visualização por parte do usuário.

Entradas		Saídas	
Alarm_Carr	<input type="checkbox"/>	Carrega	0
Alarm_Cent	<input type="checkbox"/>	Lateral_D	0
Alarm_Lat	<input type="checkbox"/>	Lateral_E	0
Alarm_Insp	<input type="checkbox"/>	Central	0
Alarm_Desc	<input type="checkbox"/>	Inspeção	0
Sens_Cod1	<input type="checkbox"/>	Descarrega	0
Sens_Cod2	<input type="checkbox"/>	Refugo	0
Sens_Cent	<input type="checkbox"/>	Cent_M/D	0
Sens_Lat	<input type="checkbox"/>	Cent_1/2	0
Sens_Insp	<input type="checkbox"/>	LatD_M/D	0
Sens_Desc	<input type="checkbox"/>	LatD_1/2	0
Sens_Curva1	<input type="checkbox"/>	LatE_M/D	0
Sens_Curva2	<input type="checkbox"/>	LatE_1/2	0
B11	<input type="checkbox"/>		
B12	<input type="checkbox"/>		
B21	<input type="checkbox"/>		
B22	<input type="checkbox"/>		
B31	<input type="checkbox"/>		
B32	<input type="checkbox"/>		
Alim_Cent	<input checked="" type="checkbox"/>		
Alim_Lat	<input checked="" type="checkbox"/>		

Figura 5.7: Estado das Entradas e Saídas do PIPEFA

Estados das Saídas				
<input type="checkbox"/> Carrega	<input type="checkbox"/> Central	<input type="checkbox"/> Refugo	<input type="checkbox"/> LatD_M/D	<input type="checkbox"/> LatE_1/2
<input type="checkbox"/> Lateral_D	<input type="checkbox"/> Inspeção	<input type="checkbox"/> Cent_M/D	<input type="checkbox"/> LatD_1/2	<input type="checkbox"/> Parar
<input type="checkbox"/> Lateral_E	<input type="checkbox"/> Descar.	<input type="checkbox"/> Cent_1/2	<input type="checkbox"/> LatE_M/D	

Figura 5.8: Estado das Saídas do Projeto PIPEFA

A listagem completa deste programa de supervisão para a plataforma PIPEFA é apresentada no Anexo IV.

5.1 Programação do CLP Ligado ao Sistema de Supervisão

Para os últimos testes da plataforma PIPEFA, dividiu-se seu funcionamento em 5 processos consecutivos, a saber:

1. Carregamento e Leitura do Código de Barras;
2. Montagem Central;
3. Montagem Lateral;
4. Inspeção;
5. Descarregamento.

A presença de base em cada um dos processos indica processo ativo, e é caracterizado pelas suas respectivas variáveis:

1. Fazer_CB;
2. Fazer_Cent;
3. Fazer_Lat;
4. Fazer_Insp;
5. Fazer_Desc.

O Sistema de Supervisão avisará o CLP sobre a utilização de um determinado posto de trabalho através da variável Fazer. Por exemplo, Fazer_Cent = 1 indica que existe uma base em processo de montagem central ou se deslocando para lá. Fazer_Cent = 0, indica que este processo já foi finalizado e o posto de trabalho está disponível.

Estas variáveis atuarão nos processos de montagem de forma direta ou indireta.

A atuação de forma direta se dá quando estas variáveis são usadas para controlar as travas do sistema, como mostrado na Tabela 5.2.

Tabela 5.2: Controle das Travas da Plataforma PIPEFA.

Processo	Variável		Atuador:
	mnemônico	Endereço	Endereço
Carregamento e Leitura do Código de Barras	Fazer_CB	C60	Y1
Montagem Central	Fazer_Cent	C61	Y2
Montagem Lateral	Fazer_Lat	C62	Y4
Inspeção	Fazer_Insp	C63	Y11
Descarregamento	Fazer_Desc	C64	Y13

A atuação de forma indireta se dá quando estas variáveis são usadas para avisar o CLP sobre a disponibilidade de utilização de um determinado posto de trabalho. Neste caso, para o CLP iniciar determinado processo ele deverá receber informações adicionais, como mostrado na Tabela 5.3.

As variáveis C50, C51, C52, C53 e C54 não constam das tabelas de variáveis do sistema de supervisão (Tabela 8.2 e Tabela 8.3) pois não são utilizadas pelo sistema de supervisão. São variáveis de interesse único e exclusivo do CLP responsável pelo processo ao qual a variável

está ligada. Por exemplo, C51 ativo indica que o CLP responsável pela montagem central deverá executá-la.

Tabela 5.3: Operação dos Postos de Trabalho Controlados pelo CLP.

Processo	Condição para início de processo		Atuação: Endereço
	Mnemônico	Endereço	
Carregamento e Leitura do Código de Barras	Carrega • Fazer_CB	C20 • C60	C50
Montagem Central	Central • Sens_Cent • Fazer_Cent	C23 • X3 • C61	C51
Montagem Lateral	Lateral_E • Sens_Lat • Fazer_Lat + Lateral_D • Sens_Lat • Fazer_Lat	C22 • X6 • C62 + C21 • X6 • C62	C52
Inspeção	Inspecao • Sens_Insp • Fazer_Insp	C24 • X13 • C63	C53
Descarregamento	Descarrega • Sens_Desc • Fazer_Desc	C25 • X16 • C64	C54

O diagrama de relés para a atuação direta das variáveis de processo é mostrado na Figura 5.9, e o diagrama de relés para atuação indireta destas variáveis é mostrado na Figura 5.10. Estes programas são executados pelo CLP ligado ao sistema de supervisão.

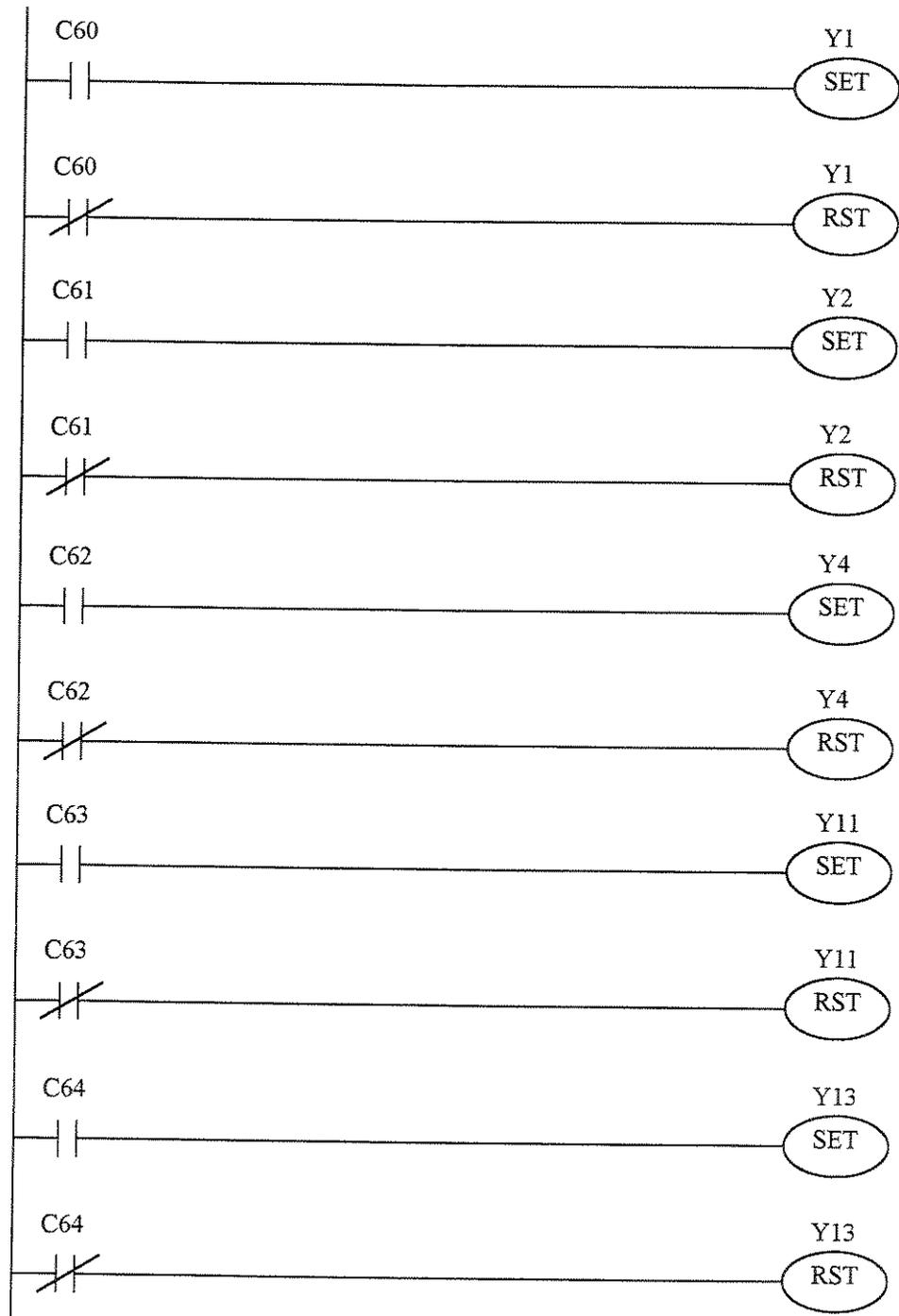


Figura 5.9: Diagrama de Relés das Travas do PIPEFA

É importante observar que o sistema de supervisão não tem uma atuação direta sobre os atuadores do sistema. Ele controla os atuadores por meio de variáveis de controle. Dessa forma, se o sistema de supervisão estiver inativo, a plataforma pode ser operada por meio da

manipulação direta dessas variáveis de controle, ou seja, mesmo que o sistema de supervisão não esteja operando pode-se utilizar a plataforma.

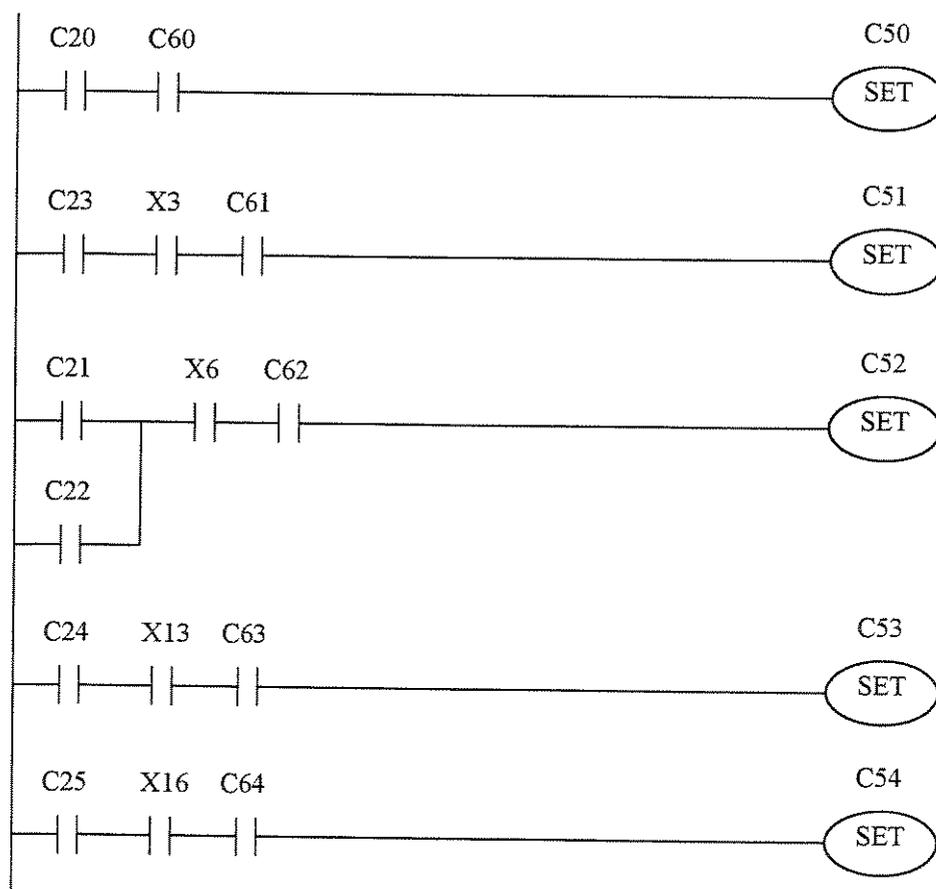


Figura 5.10: Diagrama de Relés dos Processos do PIPEFA

5.2 Teste do Sistema de Supervisão

O sistema de supervisão permite o trabalho “on line” no modo real, e o “off line” no modo de simulação. No modo real, o programa procura por informações em meios externos, e no modo de simulação, estas informações são lidas diretamente do teclado.

Inicialmente testou-se o programa no modo de simulação, utilizando-se o teclado para simular o envio de informações dos postos de trabalho. Foram testadas várias configurações de funcionamento, como montagem simultânea de oito produtos, erros de montagem, valor do primeiro código de barras diferente do segundo, intervenção de operador alterando a seqüência de montagem ou eliminando um produto, geração de alarmes e desligamento do sistema.

Na segunda bateria de testes, utilizou-se uma placa de entrada/saída e a interface paralela para a troca de informações com o meio externo. Como a parte física da plataforma ainda não estava completamente pronta, optou-se por utilizar um conjunto de interruptores para simular o funcionamento dos postos que ainda não estavam prontos. Foram também testadas várias condições de funcionamento, montagem, e feita a validação do experimento. A utilização da interface paralela, para aquisição de dados, foi feito de acordo com o proposto por Berk (1986) e está apresentada no Anexo V. A programação para utilização desta interface foi feita de acordo com Bulback (1998) e é apresentada no Anexo VI.

Na terceira e última bateria de testes, utilizou-se o CLP da Koyo para troca de informações acerca das variáveis do sistema. Utilizando-se a interface serial do computador, para comunicação com o CLP da Koyo, conseguiu-se ter acesso a todas as variáveis do sistema, monitoradas pelo CLP da Koyo. Esta interface de comunicação utiliza a porta serial do microcomputador, para fazer uma troca dinâmica de dados, em tempo real, entre os dados contidos no CLP e o microcomputador. Dessa forma, para enviar ou receber dados do CLP, deve-se executar um aplicativo que trabalhe com o protocolo DDE. A configuração desta interface é mostrada no Anexo V, e um exemplo de programação é mostrado no Anexo VI.

Nos testes realizados chegou-se a montar 5 produtos simultaneamente. Como o processo de montagem é dinâmico e automático, o número de produtos em processo de montagem variava de 3 a 5, de acordo com o tempo gasto em cada estágio de montagem.

Os pedidos de fabricação eram emitidos por outro computador, ligado em rede WindowsNT® ao computador executando o sistema de supervisão. Desta forma os pedidos de fabricação eram enviados via rede de dados ao sistema de supervisão, o qual liberava a fabricação conforme disponibilidade do sistema.

De um terceiro microcomputador, ligado aos demais pela rede WindowsNT®, conseguiu-se também monitorar o processo de montagem. Apesar deste monitoramento não ser em tempo real, é possível utilizar a opção atualizar para fazer com que o monitoramento reflita o estado atual do sistema.

A única falha do sistema de supervisão foi na leitura do código de barras. Como o sistema executa muitas operações simultaneamente, às vezes a leitura do código de barras é perdida, dando a impressão que no momento em que o leitor de código de barras fez a leitura, o sistema estava ocupado com outras operações. Como este tipo de erro é relativamente raro, cerca de 2 % das leituras realizadas, ele não comprometeu o desempenho do sistema. Além do mais, para corrigir este erro basta forçar uma nova leitura do código de barras. De qualquer forma é interessante, para trabalhos futuros, idealizar outra opção para leitura do código de barras, como por exemplo deixá-la a cargo do CLP.

A plataforma PIPEFA permaneceu em teste por horas, montando mais de uma centena de produtos sem que houvessem falhas graves do sistema de supervisão. As falhas que ocorreram foram problemas mecânicos, mostrando que a plataforma PIPEFA necessita de alguns pequenos ajustes mecânicos.

5.3 Comparação com Sistema de Supervisão Industrial

A comparação entre o sistema de supervisão, desenvolvido neste trabalho, com um desenvolvido a partir de um software comercial especializado, será feita a partir do trabalho de Martinie (1998). Martinie (1998) desenvolveu o trabalho “Structuration De La Commande Des Systemes De Production Complexes” no LAR da UNICAMP, através do convênio entre a UNICAMP e o LIISI.

No trabalho de Martinie (1998) foi utilizado um CLP TSX-37 para os postos de montagem, desmontagem, inspeção e transferência, e um CLP TSX-17 para os postos de

montagem e desmontagem, ambos da marca Telemecanique do grupo Schneider. Para a comunicação entre os CLP e o computador foi utilizada uma rede de comunicação do tipo UNITELWAY. O programa de supervisão foi desenvolvido utilizando-se a linguagem de programação P-CIM, da Afcon Inc.

Como o P-CIM possui os “drivers” para comunicação através de rede UNITELWAY, foi possível a comunicação, em tempo real, entre o computador e ambos os CLPs. Uma visão geral da interface gráfica implementada é mostrada na Figura 5.11.

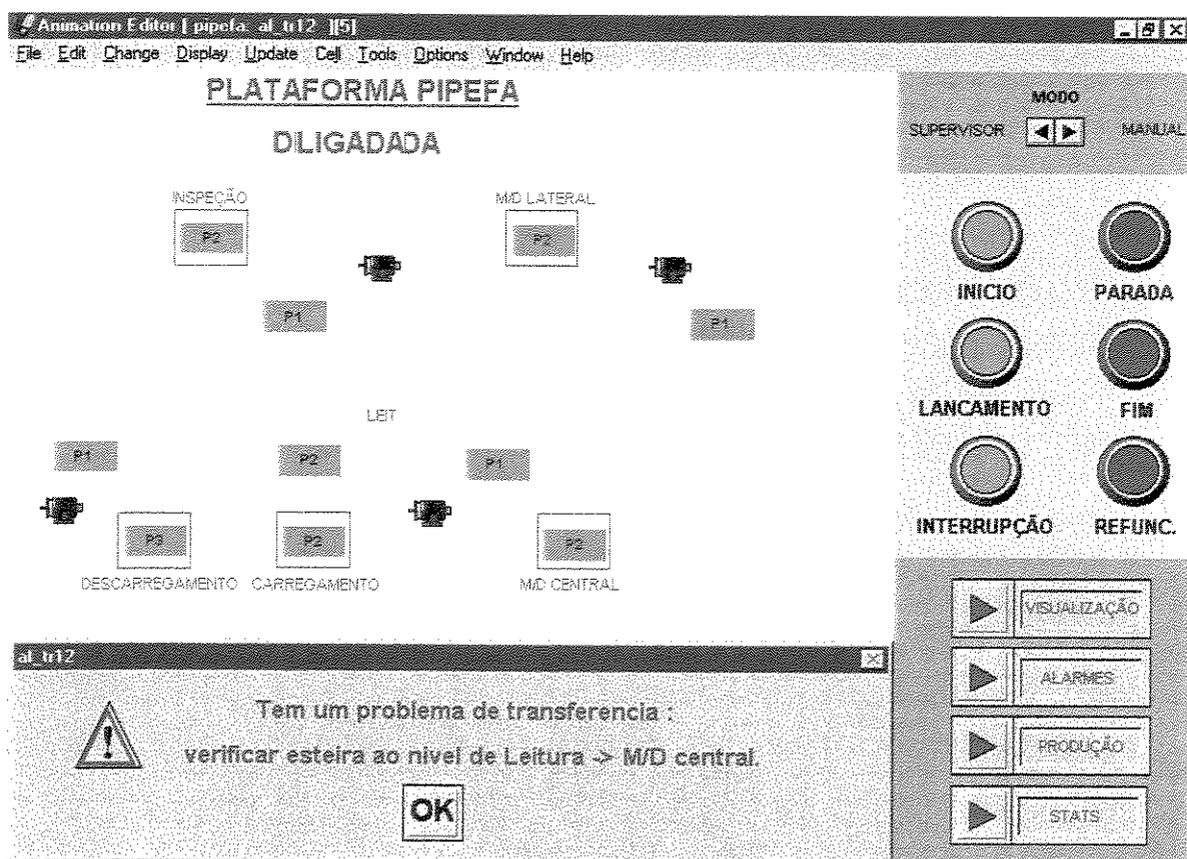


Figura 5.11: Sistema de Supervisão Desenvolvido a partir do P-CIM

No desenvolvimento do programa, foi constatada uma grande facilidade do P-CIM para geração de alarmes, facilidade esta também apresentada no Visual Basic. A geração de alarme é relativamente simples, pois é necessário apenas de um ciclo contínuo de “scanning”, verificando as variáveis e mostrando os resultados. A tela de geração de alarmes desenvolvida com o P-CIM é apresentada na Figura 5.12.

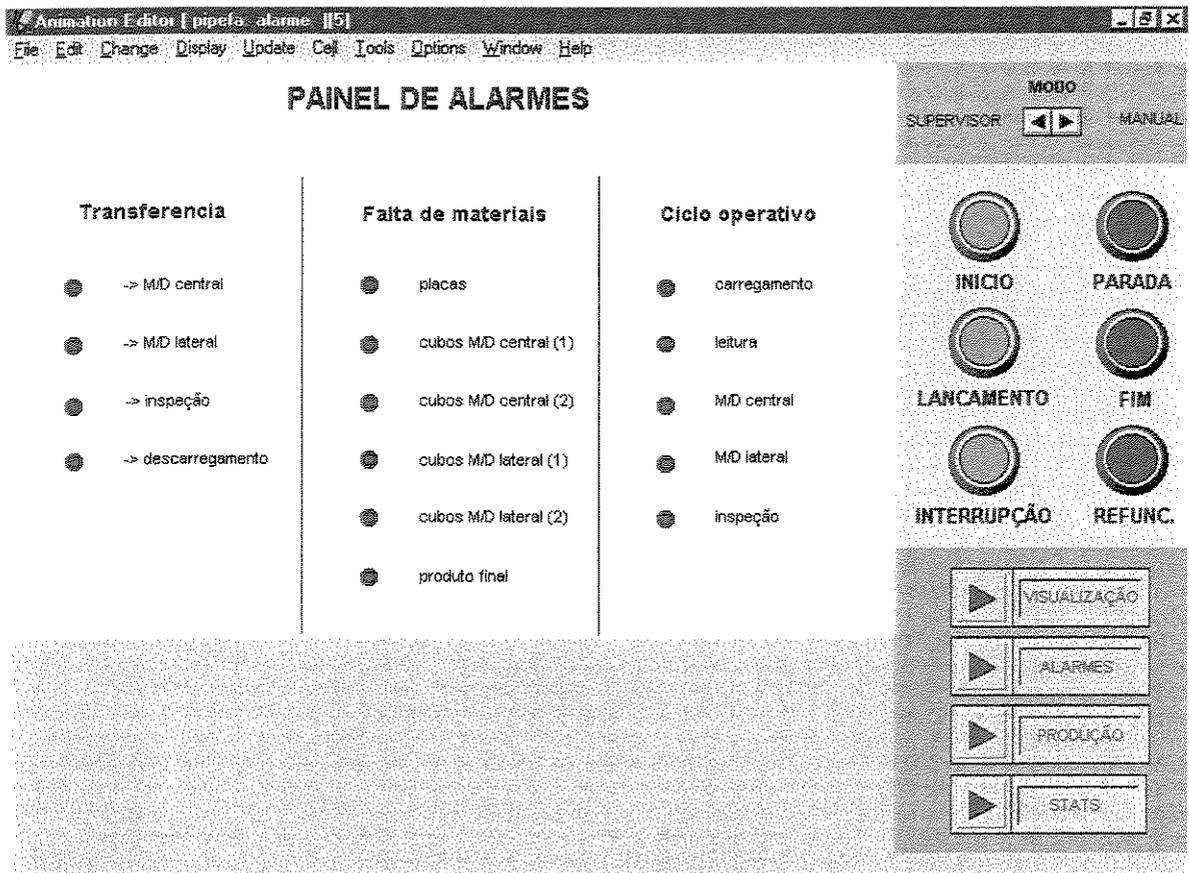


Figura 5.12: Geração de Alarmes Utilizando-se P-CIM

Infelizmente o P-CIM não possui a facilidade do Visual Basic para a utilização de banco de dados, de forma que o banco de dados utilizado armazenava informações apenas de fim de

processo. O “status” do andamento dos processos de fabricação, também não pode ter disponibilidade via Internet. A comunicação simultânea com outros CLPs, que não trabalham com UNITELWAY, também não foi possível. De qualquer forma, o ambiente de programação P-CIM se mostrou prático e seguro. Os aplicativos foram de fácil desenvolvimento e não houve problemas de falhas de software, como travamento inesperado.

A programação dos CLPs TSX foi feita utilizando-se o software PL7 Micro, desenvolvido pela Schneider Automation. Este gera os programas, em Diagrama de Relés, a partir do pós-processamento de programas em SFC. Um exemplo da utilização do software PL7 Micro é mostrada na Figura 5.13.

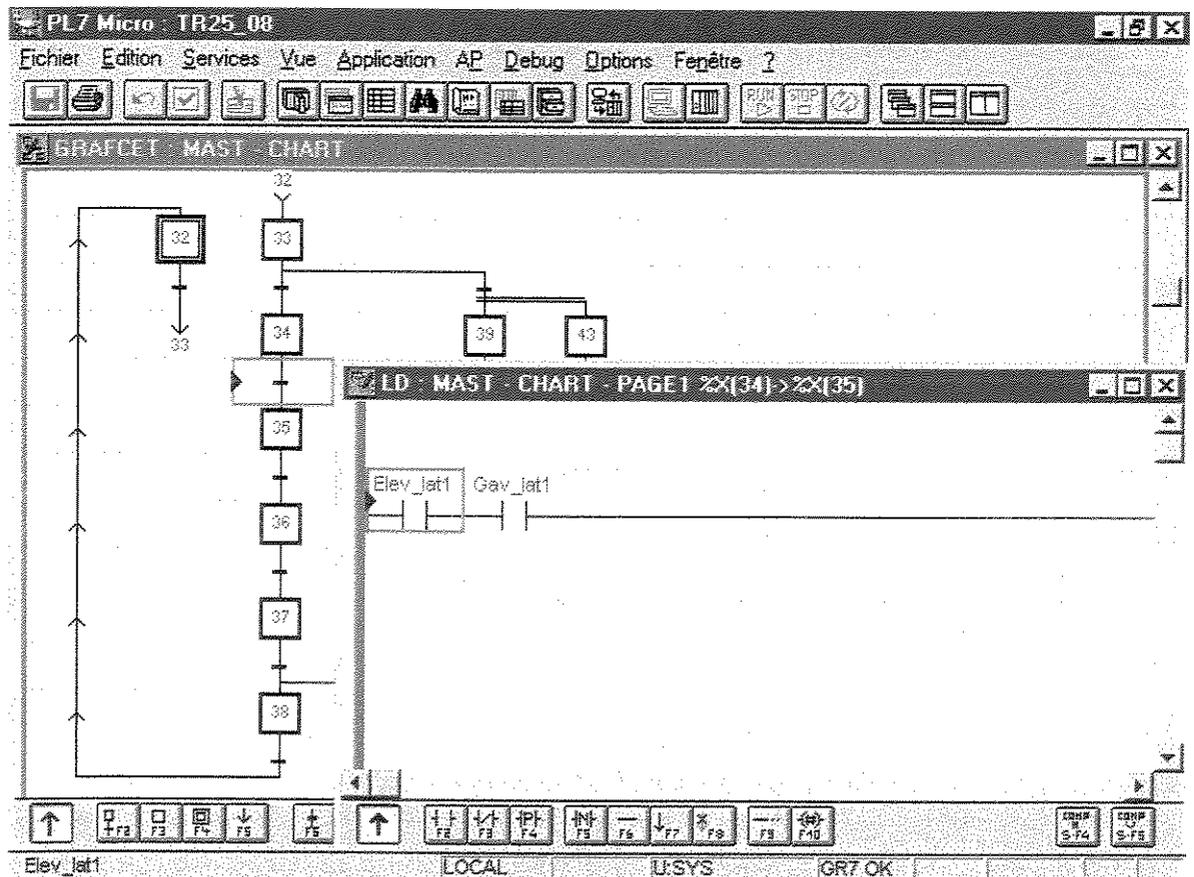


Figura 5.13: Exemplo de Utilização do PL7 Micro

Na comparação entre os sistemas de supervisão, o desenvolvido em P-CIM e o desenvolvido em Visual BASIC, notou-se que o Visual BASIC apresenta maior versatilidade e flexibilidade, pois trabalha de forma integrada com aplicativos de outros fabricantes, enquanto o P-CIM apresenta maior segurança e robustez, pois o Visual Basic apresentou algumas instabilidades quando trabalhando com mais de 35 variáveis de Entrada/Saída. A utilização da opção de simulação gráfica do sistema de supervisão, permitiu testar o programa antes de executar as implementações físicas, propiciando uma economia de tempo e facilidade na implementação.

Neste capítulo foram apresentadas as fases de implementação e validação do sistema de supervisão aplicado ao projeto PIPEFA. Foi mostrada a importância da utilização de sistemas abertos para desenvolvimento de aplicações, deixando como forma inadequada de implementação, aquelas específicas e rígidas. Foi vista a importância dos protocolos de comunicação, na integração dos elementos constituintes do sistema automatizado de produção.

No capítulo seguinte são apresentadas as conclusões e perspectivas futuras.

Capítulo 6

Conclusões e Perspectivas Futuras

Com a realização destes projetos pode-se desenvolver uma nova técnica para programação de eventos discretos utilizando o software Visual Basic. Demonstrou-se que é possível fazer o desenvolvimento de sistema de supervisão utilizando-se uma linguagem de programação de baixo custo e amplamente difundida pelo mercado.

Foi demonstrada a enorme facilidade de programação em Diagrama Seqüencial, se comparado ao tradicional Diagrama de Relés.

Com este projeto foi possível desenvolver e implementar uma estrutura de aprendizagem e validação de programas para Sistemas Automatizados de Produção. Utilizando-se animação gráfica e desenvolvimento de interfaces homem-máquina, conseguiu-se desenvolver e implementar um software educativo aplicado à estrutura SAP.

A utilização de banco de dados SQL para controle de produção mostrou-se uma excelente alternativa. Trabalhando com rapidez e segurança, permitiu estabelecer um fluxo contínuo de informações entre as camadas do CIM.

Este projeto permitiu estabelecer uma definição da arquitetura do sistema de supervisão, cuja estrutura poderá ser utilizada em outros projetos ligados à plataforma PIPEFA.

Este projeto demonstrou a grande importância dos protocolos de comunicação entre os níveis e intra-níveis da pirâmide CIM. Também foi demonstrado a necessidade de padronização desses protocolos.

A utilização de sistemas abertos mostrou-se uma excelente solução para desenvolvimento de programas de supervisão, monitoração e controle, pois graças a sua adaptação a diversas formas de aplicação, sua aceitação de novas tecnologias, sua atualização e sua independência é possível garantir uma melhor conectividade e interoperabilidade, aumentando assim a flexibilidade do sistema.

Este trabalho viabiliza trabalhos futuros, como por exemplo, implementar o protocolo Fieldbus, ou outro protocolo que venha a se tornar padrão de comunicação para barramento, melhorando a comunicação dentro do projeto.

Finalizando, vale à pena acrescentar que, com este projeto foi criado um ambiente de desenvolvimento, ponto importante para que, trabalhos futuros, possam desenvolver aplicação de técnicas em Gestão de Produção e Controle de Qualidade.

7. Referências Bibliográficas

- Aguiar, André Luís Santos de. *Modelo de Referência da Arquitetura OSI*.
<http://www.geocities.com/SiliconValley/Horizon/4761/rm-osi.html>, 1997.
- Anderson, Peter H. Use of a PC Printer Port for Control and Data Acquisition. In: *Electronic Journal for Engineering Technology*, Fall 1996. Department of Electrical Engineering. Morgan State University. <http://et.nmsu.edu/~etti/fall96/computer/printer/printer.html>.
- Becerra, J. L. R. *Especificação, Projeto e Implementação de um Controlador de Célula Flexível Aplicado na Manufatura*. São Paulo: POLI, USP, 1993. 166p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo, 1993.
- Berk, A. A. *Microcontrollers in process and product control*. EUA : McGraw-Hill, 1986. 211p.
- Bittar, R. C. S. M. *A Utilização do Grafcet como Ferramenta na Automação Industrial*. Campinas: FEM, UNICAMP, 1993. 105p. Dissertação (Mestrado) – Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1993.
- Bolton, W. *Mechatronics: electronic control systems in mechanical engineering*. London : Longman, c1995. 380p.
- Brendel, W.; Tiegelkamp, M. *Uniform PLC programming in accordance with IEC 1131-3*. Germany : International Electrotechnical Commission, 1992.
- Bulback, Fred. *Fred Bulback's Homepage*. <http://come.to/fredb> ou <http://www.silk.net/personal/fredb/menu.html>, 1998.
- David, R. Grafcet: a powerful tool for specification of logic controllers. In: *IEE Transactions on*

- control systems technology*, v. 3, n. 3, p. 253-268, Setembro, 1995.
- DiCesare, F.; Harhalakis, G.; Proth, J. M.; Silva M.; Vemadat, F. B. *Practice of Petri nets in manufacturing*. London : Chapman & Hall, 1993. 295p.
- FANUC. *Manual del Mantenimiento*. Japan, abril. 1990. Sp.p (Fanuc Series 0/00/0-Mate).
- Fanuc. *Manual del Operador*. Japan, jun. 1991. Sp.p (Fanuc Series 0-MC / 00-MC / 0-Mate-MC).
- Ferreira, A. B. H. *Novo Dicionário da Língua Portuguesa*. 2. ed. Rio de Janeiro, Nova Fronteira, 1986.
- GE Fanuc Automation. *Manual del Mantenimiento*. Equipos de Control Numerico. Japão, 1988. Sp.p (GE Fanuc Serie 0/00/0-Mate).
- GE Fanuc Automation. *Manual del Operador*. Equipos de Control Numerico. Japão, 1988. Sp.p (GE Fanuc Serie 0-TC/00-TC/0-Mate TC).
- Guia do Programador. *Sistema de programação para Windows95 e WindowsNT, versão 5.0*. Microsoft Visual Basic. 1997. p. 5 – 7.
- Hart, Matt. Matt Hart's Visual Basic Help Page. <http://www.matthart.com/vbhelp/>, 1999
- Industrial Control and System. *Programmable Controllers*. National Electrical Manufacturers Association. part. 3-304, p.38-48, 1988.
- Intel Corporation. *USB 1.0 Spec Position on Extension Cables and Pass-Through Monitors*. White Paper. <http://www.usb.org/developers/index.html>, 1998.
- International Electrotechnical Commission. *Programmable Controllers*. IEC Standard 1131-3, 1993.
- Jafari, M. A. Supervisory control specification and synthesis. In: Zhou, M., (ed.) *Petri Nets in*

- flexible and agile automation*. Boston, Kluwer Academic Publishers, 1995. cap. 12.
- Martinie, Luc. *Structuration De La Commande Des Systemes De Production Complexes*. São Paulo, UNICAMP, 1998.
- Murata, T. Application of Petri Nets to sequence control programming. In: Zhou, M., (ed.) *Petri Nets in flexible and agile automation*. Boston, Kluwer Academic Publishers, 1995. cap. 2.
- Peluso, M. Como o Fieldbus Afetará o seu Próximo Projeto. *Controle & Instrumentação*, p. 11-12, Janeiro/Fevereiro, 1997.
- ROMI. *Diagramas Elétricos*. Polaris Mach 6. Brasil, maio, 1995. sp.p.
- ROMI. *Manual de Manutenção Eletrônica*. Sistema Fanuc. Brasil, junho, 1993. 142 p.
- Rosário, J. M., Frachet, J. P. *Métodos e Ferramentas em Automação Industrial e Produção para o Desenvolvimento da Qualidade e da Produtividade nas PME/PMI*. Projeto de Pesquisa, CAPES/COFECUB, Junho, 1996.
- Rosário, J. M. O Grafcet na Concepção de Sistemas Automatizados. *Schneider News*. São Paulo, ano 1, n.2, p. 11, Maio 1997.
- Rosário, J. M, Fauré, J. M. Um Exemple de Coopération Franco-Brésilienne pour la Formation et la Recherche en Automatisation Industrielle. In Troisième Conférence Internationale Sur L'Automatisation Industrielle, *Third International Conference on Industrial Automation*. Montreal, junho, 1999.
- Santos, P. C. F. Delphi x Visual Basic. *Byte Brasil*. v. 06, n. 07, p. 24 – 28, Julho, 1997.
- Silveira, Paulo R. da, Santos, Winderson E. *Automação e Controle Discreto*. São Paulo: Érica, 1998. 229p.
- Scheer, A.-W. *CIM: Computer Integrated Manufacturing: Towards the Factory of the Future*. 3. ed. Germany : Springer-Verlag, 1994. 303p.

Souza, Libânio C.; Peluso, Marcos *Tutorial Fieldbus – SMAR*, Divisão de Pesquisa e Desenvolvimento, 1996. 8 p.

Synergetic Micro Systems, Inc. *Fieldbus Comparison (PROFIBUS, InterBus, CAN...)*.
<http://www.synergetic.com/compare.htm> , 1997.

Tanenbaum, A. S. *Redes de computadores*. 2.ed. Tradução de: *Computer Networks*. Rio de Janeiro: Campus, c1994. 786p.

Teicholz, E., Orr, J. N. *Computer Integrated Manufacturing Handbook*. Singapura: McGraw-Hill, 1989. 444p.

TELEMEC. *O Grafcet: Diagrama funcional para automatismos seqüenciais*. Telemecanique. Nov. 1986.

Thompson, Lawrence M. *Industrial Data Communications*. EUA : Instrument Society of America, 1991. 217p.

8. Anexos

Anexo I – Possíveis Configurações de Montagem

Tabela 8.1: Especificação dos Códigos Decimal e Base três

Deci.	Visualização	B. 3
0		000
1		001
2		002
3		010
4		011
5		012
6		020
7		021
8		022

Deci.	Visualização	B. 3
9		100
10		101
11		102
12		110
13		111
14		112
15		120
16		121
17		122

Deci.	Visualização	B. 3
18		200
19		201
20		202
21		210
22		211
23		212
24		220
25		221
26		222

Anexo II – Relação das Variáveis de Entrada/Saída da Plataforma PIPEFA

Tabela 8.2: Relação de Entradas (CLP-PC)

Régua	Variáveis	Ender.	CLP	Descrição Geral
1	SENS_COD1	X2	PMDL	Sensor de chegada no leitor de código de barras 1
2	SENS-CENT	X3	PMDC	Sensor de chegada de placa no posto M/D central
3	SENS-LAT	X6	PMDL	Sensor de chegada de placa no posto M/D lateral
4	SENS-INSP	X13	PI	Sensor de chegada do produto no posto de Inspeção
5	SENS-DESC	X16	PCD	Sensor de chegada do produto no posto de Descarregamento
6	SENS_CURVA1	X21	PMDL	Sensor de passagem de placas antes de iniciar a curva 1
7	SENS_CURVA2	X22	PI	Sensor de passagem de placas antes de iniciar a curva 2
8	ELEV_CENT	X4		Sensor do elevador do posto de M/D Central recuado
9	ELEV_LAT	X7		Sensor do elevador do posto de M/D Lateral recuado
10	ELEV_INSP	X14		Sensor do elevador do posto de Inspeção recuado
11	B11	C10	PI	Código do Produto (posição B11)
12	B12	C11	PI	Código do Produto (posição B12)
13	B21	C12	PI	Código do Produto (posição B21)
14	B22	C13	PI	Código do Produto (posição B22)
15	B31	C14	PI	Código do Produto (posição B31)
16	B32	C15	PI	Código do Produto (posição B32)
17	ALARM_CARR	C0	PCD	Alarme de mal funcionamento do Posto de Carregamento
18	ALARM_CENT	C1	PMDC	Alarme de mal funcionamento do Posto Montagem Central
19	ALARM_LAT	C2	PMDL	Alarme de mal funcionamento do Posto Montagem Lateral
20	ALARM_INSP	C3	PI	Alarme de mal funcionamento do Posto de Inspeção
21	ALARM_DESC	C4	PCD	Alarme de mal funcionamento do Posto de Descarregamento
22	ALIM_CENT		PMDC	Alimentação de cubos no carregador (posto M/D central)

Tabela 8.3: Relação de Saídas (PC_CLP)

Régua	Variáveis	Ender.	CLP	Descrição Geral
1	CARREGA	C20	PCD	Carregamento de placa de base no sist. de transferência
2	LATERAL_D	C21	ST	Informação de utilização do posto M/D lateral direita
3	LATERAL_E	C22	ST	Informação de utilização do posto M/D lateral esquerda
4	CENTRAL	C23	PMDC	Informação de utilização do posto M/D central
5	INSPECAO	C24	PI	Informação de utilização do posto de Inspeção
6	DESCARREGA	C25	PCD	Informação de utilização do posto de descarregamento
7	REFUGO	C26	PI	Ordem para rejeição do Produto (produto descartado)
8	CENT_M/D	C27	PMDC	Operação de Montagem/Desmontagem (central)
9	CENT_1/2	C30	PMDC	Montagem de 1 ou 2 cubos (central)
10	LATD_M/D	C31	PMDL	Operação de Montagem/Desmontagem (lateral direita)
11	LATD_1/2	C32	PMDL	Montagem de 1 ou 2 cubos (lateral direita)
12	LATE_M/D	C33	PMDL	Operação de Montagem/Desmontagem (lateral esquerda)
13	LATE_1/2	C34	PMDL	Montagem de 1 ou 2 cubos (lateral esquerda)
14	FAZER_CB	C60		Habilita a leitura do Código de Barras
15	FAZER_CENT	C61		Habilita a utilização do Posto de M/D Central
16	FAZER_LAT	C62		Habilita a utilização do Posto de M/D Lateral
17	FAZER_INSP	C63		Habilita a utilização do Posto de Inspeção
18	FAZER_DESC	C64		Habilita a utilização do Posto de Descarregamento

Anexo III – Grafset do Sistema de Supervisão para o PIPEFA

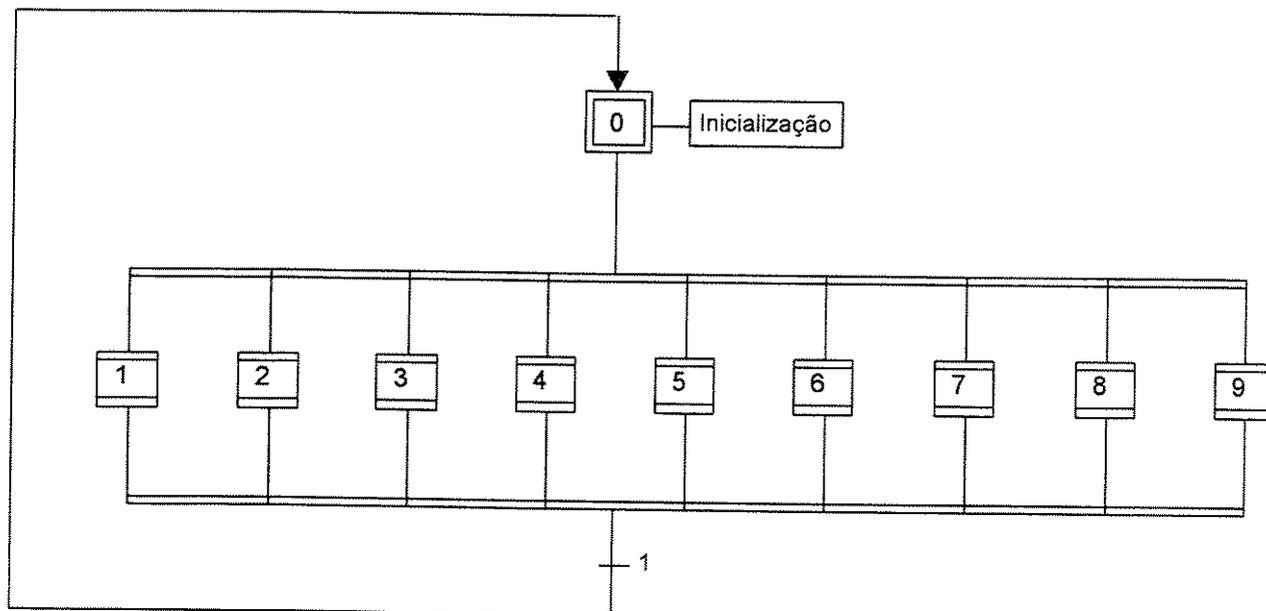
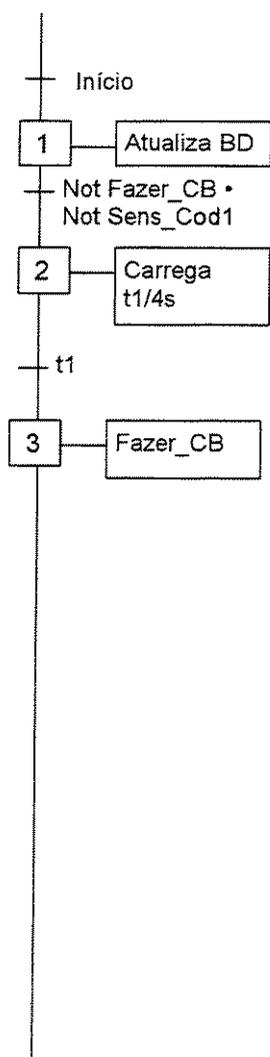
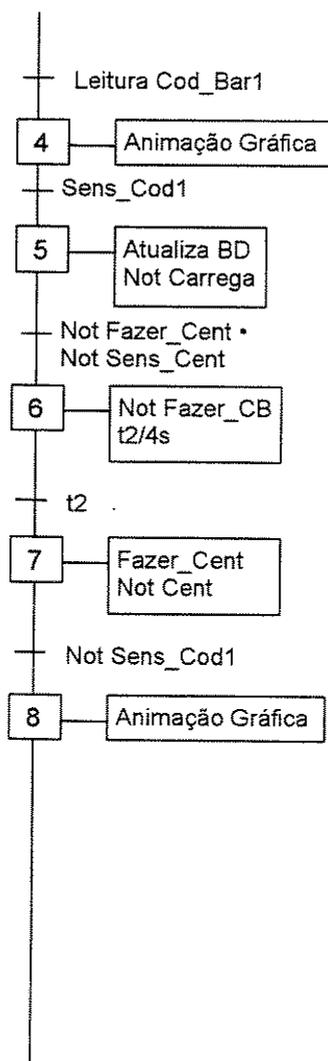


Figura 8.1: Grafset do Sistema de Supervisão para o PIPEFA

Macro 1:
Carregamento



Macro 2:
Código Barras I



Macro 3:
Montagem Central

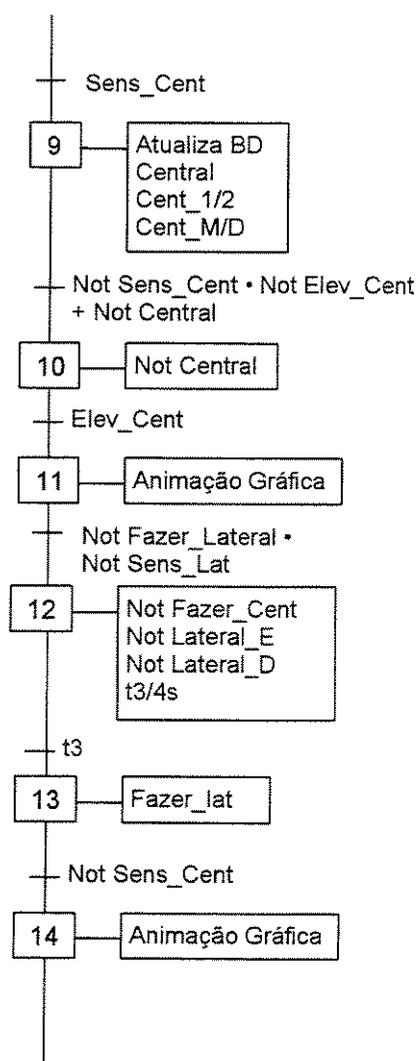
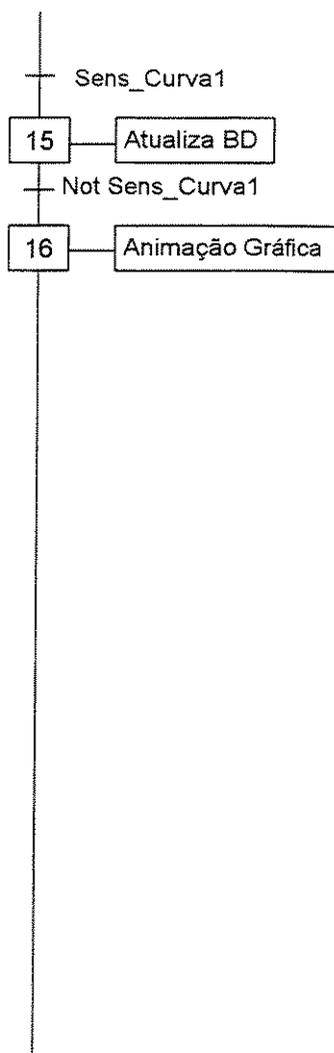
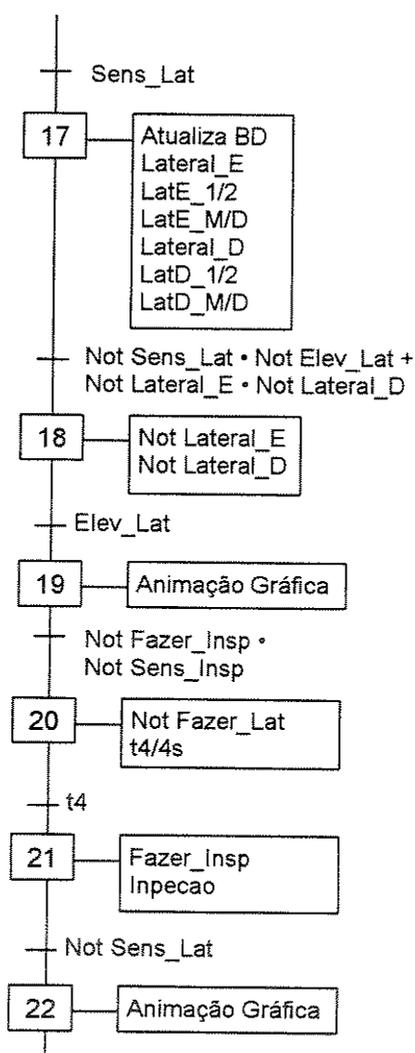


Figura 8.2: Macros 1, 2 e 3 do Grafcet PIPEFA

Macro 4:
Curva 1



Macro 5:
Montagem Lateral



Macro 6:
Código Barras II

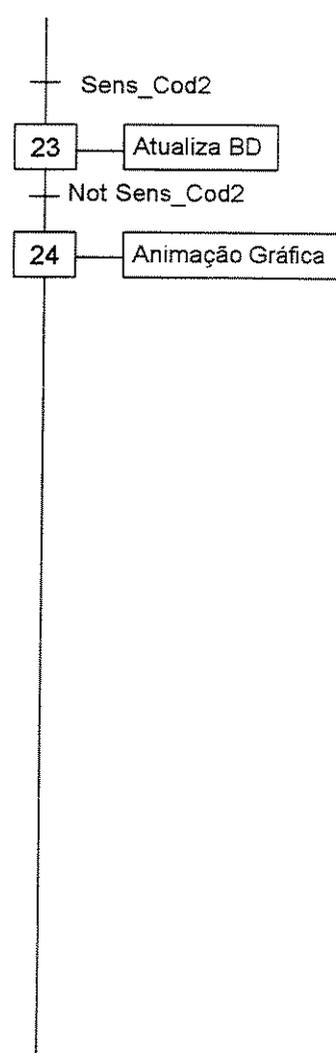


Figura 8.3: Macros 4, 5 e 6 do Grafcet PIPEFA

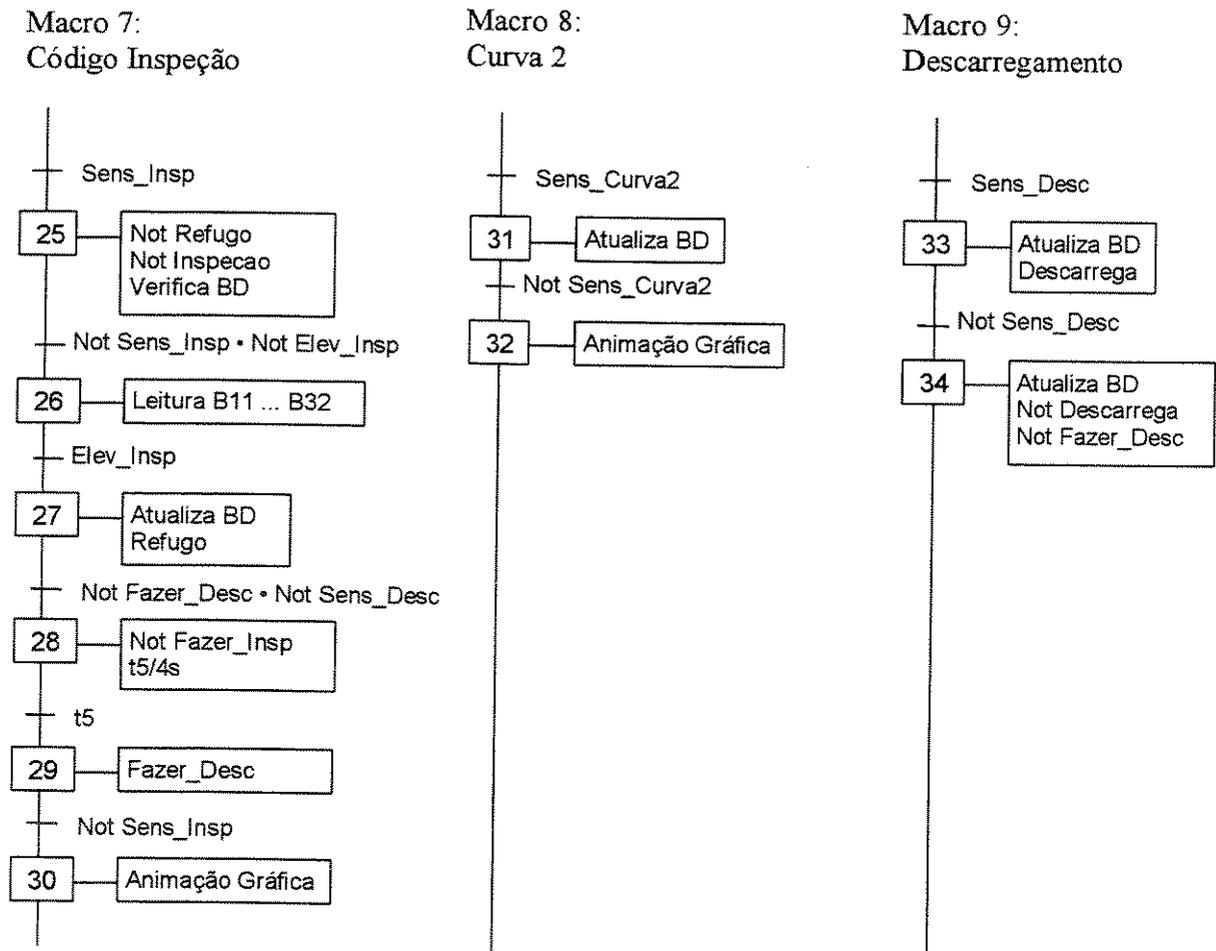


Figura 8.4: Macros 7, 8 e 9 para o Grafcet PIPEFA

Anexo IV – Programa de Supervisão para o PIPEFA

Este anexo apresenta o formulário principal do programa de Supervisão para o projeto PIPEFA. Ele foi desenvolvido em Visual Basic, e a parte mais importante deste formulário é a rotina Private Sub Timer1_Timer(). Esta rotina executa um “scanning” contínuo do sistema, sendo responsável por todo o seu funcionamento.

```
Private Sub B_KeyPress(Index As Integer, KeyAscii As Integer)
    nt = KeyAscii - 64
End Sub
```

```
Private Sub codigo1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then 'Esperar por Enter
'If KeyAscii = 32 Then 'Esperar por espaço
    'Irá armazenar os três últimos dígitos do valor
    'da entrada na variável auxiliar aux1
    aux1 = Val(Codigo1.Text)
    If aux1 > 99 Then 'selecionar os 2 últimos dígitos
        Codigo1.Text = Mid$(Codigo1.Text, Len(Codigo1.Text) - 1, 2)
        aux1 = Val(Codigo1.Text)
    End If
    Codigo1.SelStart = 2 'cursor no final da linha
    Barcod1 = aux1
    Codigo1.Enabled = False
    Platafor.Plataforma.SetFocus
    Logica2 = Logica2 + 1
End If
End Sub
```

```
Private Sub codigo2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then 'Esperar por Enter
    'Irá armazenar os três últimos dígitos do valor
    'da entrada na variável auxiliar aux1
    aux1 = Val(Codigo2.Text)
    If aux1 > 99 Then 'selecionar os 2 últimos dígitos
        Codigo2.Text = Mid$(Codigo2.Text, Len(Codigo2.Text) - 1, 2)
```

```

        aux1 = Val(Codigo2.Text)
    End If
    Codigo2.SelStart = 2 'cursor no final da linha
    Barcod2 = aux1
    Platafor.Plataforma.SetFocus
    Logica6 = Logica6 + 1
End If
End Sub

Private Sub Controle_KeyPress(KeyAscii As Integer)
    nt = KeyAscii - 64
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    nt = KeyAscii - 64
End Sub

Private Sub Form_Load()
' ItemAbrir_Click
    ItemRedesenhar_Click
    ItemTestar_Click
End Sub

Private Sub InsCodMont_Click()
    Pedidos.Show
End Sub

Private Sub InsCodMont_KeyPress(KeyAscii As Integer)
    nt = KeyAscii - 64
End Sub

Private Sub ItemAbrir_Click()
    On Error Resume Next
    CommonDialog1.filename = "Superv2.mdb"
    CommonDialog1.CancelError = -1
    CommonDialog1.Filter = "Base de dados | *.mdb"
'CommonDialog1.Flags = cdlOFNReadOnly
    CommonDialog1.Action = 1
    If Err = 0 Then
        FileCopy Arquivo, CommonDialog1.filename
        DataBanc2.datPrimaryRS.DatabaseName = CommonDialog1.filename
    End If
End Sub

```

```

    DataBanc2.datPrimaryRS.RecordSource = "Tabela1"
    DataBanc2.datPrimaryRS.Refresh ' Open the Database.
    Plataforma.Dados.DatabaseName = CommonDialog1.filename
    Plataforma.Dados.RecordSource = "Tabela1"
    Plataforma.Dados.Refresh 'Atualiza o Bando de Dados
End If
End Sub

Private Sub ItemBancoDados_Click()
' DataBanc2.Data1.Refresh
    DataBanc2.Show
End Sub

Private Sub ItemDeletar_Click()
    'Deletar o banco de dados anterior
    Dados.Refresh
    If Not Dados.Recordset.EOF Then Dados.Recordset.MoveFirst
    Do While Dados.Recordset.EOF = False
        Dados.Recordset.Delete
        Dados.Recordset.MoveNext
    Loop
    Plataforma.Dados.Refresh
    Platafor.Plataforma.SetFocus
    ItemRedesenhar_Click
End Sub

Private Sub ItemSaidas_Click()
    Saidas.Show
    Entra_Sai.Show
End Sub

Private Sub ItemRedesenhar_Click()
    On Error GoTo TrataErro
    'Desativar figuras
    For i = 0 To 18
        Estágio(i).Visible = False
    Next i
    For i = 0 To 5
        B(i).Visible = False
        BL(i).Visible = False
        BLine(i).Visible = False
    Next i
End Sub

```

```
Next i
Mensagem.Caption = "Painel de Mensagens"
Logica1 = 0
Logica2 = 0
Logica3 = 0
Logica4 = 0
Logica5 = 0
Logica6 = 0
Logica7 = 0
Logica8 = 0
Logica9 = 0
Logica10 = 0
Codigo1.Enabled = False
Central.Visible = False
LateralD.Visible = False
LateralE.Visible = False
Codigo2.Enabled = False
Image1.Visible = False
Image2.Visible = False
Image3.Visible = False
'Zerar variáveis de saída
CARREGA = False
LATERAL_D = False
LATERAL_E = False
CENT = False
INSPECAO = False
DESCARREGA = False
REFUGO = False
CENT_MD = False
CENT_12 = False
LATD_MD = False
LATD_12 = False
LATE_MD = False
LATE_12 = False
Fazer_CB = False
Fazer_Cent = False
Fazer_Lat = False
Fazer_Insp = False
Fazer_Desc = False
'Procura no banco de dados
Dados.Refresh
```

```

Dados.Recordset.MoveFirst
Do While Dados.Recordset.EOF = False
  'Verificar qual o último estágio realizado
  For i = 1 To 10
    Est = "Est" & Format(i)
    Tempo = Dados.Recordset(Est)
    If Tempo = 0 Then Exit For
  Next i
  Estág = i - 1
  CodMont = Dados.Recordset("CodMont")
  CodBar1 = Dados.Recordset("CodBar1")
  CodBar2 = Dados.Recordset("CodBar2")
  Select Case Estág
    Case 1
      Estágio(1).Visible = True
      Logical = 2
    Case 2
      'CodigoBarras1
      Estágio(2).Visible = True
      Logica2 = 4
      CARREGA = False
      Fazer_CB = True
    Case 3
      'Montagem Central
      Estágio(3).Visible = True
      Logica3 = 2
      Fazer_Cent = True
      'Leitura do Código de Montagem
      CodigoDeMontagem (CodMont)
      'Carregar as imagens
      Figura = "0" & Tb(1) & "0" & "f"
      Image1.Picture = ImageList1.ListImages(Figura).Picture
      Image1.Visible = False
      'Configurando os sinais de saída p/ posto Central
      If Tb(1) = 1 Then
        CENT = True
        CENT_12 = False
      ElseIf Tb(1) = 2 Then
        CENT = True
        CENT_12 = True
      Else

```

```

    CENT = False
End If
CENT_MD = Montag
Central.Visible = True
FigCentral = "Cubos" + Format(Tb(1))
Central.Picture = ImageList2.ListImages(FigCentral).Picture
Case 4
'Lógica intermediária 3-5
Estágio(4).Visible = True
Logica4 = 2
Fazer_Lat = True
'Leitura do Código de Montagem
CodigoDeMontagem (CodMont)
'Carregar as imagens
Figura = "0" & Tb(1) & "0" & "f"
Estágio(4).Picture = ImageList1.ListImages(Figura).Picture
Case 5 To 6
'Montagem lateral
Estágio(5).Visible = True
Logica5 = 2
Fazer_Lat = True
'Leitura do Código de Montagem
CodigoDeMontagem (CodMont)
'Carregar as figuras
Figura = Tb(2) & Tb(1) & Tb(0) & "f"
Image2.Picture = ImageList1.ListImages(Figura).Picture
Image2.Visible = False
LateralE.Visible = True
LateralD.Visible = True
FigLatE = "Cubos" + Format(Tb(2))
FigLatD = "Cubos" + Format(Tb(0))
LateralE.Picture = ImageList2.ListImages(FigLatE).Picture
LateralD.Picture = ImageList2.ListImages(FigLatD).Picture
Figura = "0" & Tb(1) & "0" & "f"
Estágio(5).Picture = ImageList1.ListImages(Figura).Picture
'Configurando os sinais de saída da Lat. Esquerda
If Tb(2) = 1 Then
    LATERAL_E = True
    LATE_12 = False
ElseIf Tb(2) = 2 Then
    LATERAL_E = True

```

```

    LATE_12 = True
Else
    LATERAL_E = False
End If
LATE_MD = Montag
'Configurando os sinais de saída da Lat. Direita
If Tb(0) = 1 Then
    LATERAL_D = True
    LATD_12 = False
ElseIf Tb(0) = 2 Then
    LATERAL_D = True
    LATD_12 = True
Else
    LATERAL_D = False
End If
LATD_MD = Montag
Case 7
'Inspeção
Estágio(7).Visible = True
Logica7 = 6
Fazer_Insp = True
REFUGO = False
INSPECAO = False
'Leitura do Código de Montagem
CodigoDeMontagem (CodMont)
'Carregar as figuras
Figura = Tb(2) & Tb(1) & Tb(0) & "f"
Estágio(7).Picture = ImageList1.ListImages(Figura).Picture
Case 8
'Lógica intermediária
Estágio(8).Visible = True
Logica8 = 2
Fazer_Desc = True
'Leitura do Código de Montagem
CodigoDeMontagem (CodInsp)
'Carregar as figuras
Figura = Tb(0) & Tb(1) & Tb(2) & "f"
Estágio(8).Picture = ImageList1.ListImages(Figura).Picture
Case 9
'Descarga
Estágio(9).Visible = True

```

```

    Logica9 = 2
    Fazer_Desc = True
    'Leitura do Código de Montagem
    CodigoDeMontagem (CodInsp)
    'Carregar as imagens
    Figura = Tb(0) & Tb(1) & Tb(2) & "f"
    Estágio(9).Picture = ImageList1.ListImages(Figura).Picture
    DESCARREGA = True
Case 10
    'Leitura do Código de Montagem
    CodigoDeMontagem (CodMont)
    'Carregar as imagens
    Figura = Tb(0) & Tb(1) & Tb(2) & "f"
    Estágio(10).Picture = ImageList1.ListImages(Figura).Picture
    Estágio(10).Visible = True
    Logica10 = 2
End Select
Dados.Recordset.MoveNext
Loop

```

TrataErro:

```

Select Case Err
Case 0 'Falso Erro
    Exit Sub
Case 3021 'arquivo vazio
    Exit Sub
Case 53 'arquivo inexistente
    'Procurar por Banco de Dados
    ItemAbrir_Click
    Exit Sub
Case 340 'Variável inexistente
    'Procurar por Banco de Dados
    ItemAbrir_Click
    Exit Sub
Case 3024 'Banco de Dados inexistente
    'Procurar por Banco de Dados
    ItemAbrir_Click
    Exit Sub
Case Else 'Erro desconhecido
    MsgBox "Erro ao redesenhar a tela", 48, "Erro desconhecido:" & Err
    Exit Sub

```

```
End Select
Resume Next
End Sub
```

```
Private Sub ItemSair_Click()
End
End Sub
```

```
Private Sub ItemTestar_Click()
If Testar Then
Testar = False
ItemTestar.Caption = "Modo de Teste"
For i = 1 To 18
Entra_Sai.Saida(i).Caption = -Bout(i)
Entra_Sai.Saida(i).LinkMode = 2
Entra_Sai.Saida(i).LinkPoke
Next i
For i = 1 To 16
Entra_Sai.Entrada(i).LinkMode = 1
Next i
Else
Testar = True
ItemTestar.Caption = "Modo Real"
For i = 1 To 18
Entra_Sai.Saida(i).LinkMode = 0
Next i
For i = 1 To 16
Entra_Sai.Entrada(i).LinkMode = 0
Next i
End If
End Sub
```

```
Private Sub Liberar_Click()
If Logical = 0 Then
Logical = 1
Else
Logical = 0
End If
End Sub
```

```
Private Sub Liberar_KeyPress(KeyAscii As Integer)
```

```

    nt = KeyAscii - 64
End Sub

Private Sub Timer1_Timer()
Leitura
'Tradução das Entradas
    S_COD1 = Bin(1) 'Sensor Magnético 1
    S_CENT = Bin(2) 'Sensor Magnético 2
    S_LAT = Bin(3) 'Sensor Magnético 4
    S_INSP = Bin(4) 'Sensor Magnético 6
    S_DESC = Bin(5) 'Sensor Magnético 8
    S_CURVA1 = Bin(6) 'Sensor Magnético 3
    S_CURVA2 = Bin(7) 'Sensor Magnético 5
    Elev_Cent = Bin(8) 'Sensor elevador recuado
    Elev_Lat = Bin(9) 'Sensor elevador recuado
    Elev_Insp = Bin(10) 'Sensor elevador recuado
    B11 = Bin(11) 'Código de Inspeção
    B12 = Bin(12) 'Código de Inspeção
    B21 = Bin(13) 'Código de Inspeção
    B22 = Bin(14) 'Código de Inspeção
    B31 = Bin(15) 'Código de Inspeção
    B32 = Bin(16) 'Código de Inspeção
' Alarm_Car = Bin(17) 'Alarme do P. Carregamento
' Alarm_Cent = Bin(18) 'Alarme do P. Mont. Central
' Alarm_Lat = Bin(19) 'Alarme do P. Mont. Lateral
' Alarm_Insp = Bin(20) 'Alarme do P. Inspeção
' Alarm_Desc = Bin(21) 'Alarme do P. Descarregamento

'Verificação dos alarmes
    AlarmCar.Visible = Alarm_Car
    AlarmCent.Visible = Alarm_Cent
    AlarmLat.Visible = Alarm_Lat
    AlarmInsp.Visible = Alarm_Insp
    AlarmDesc.Visible = Alarm_Desc
'Processo de cintilação
    AlarmCent.FillColor = AlarmCar.FillColor
    AlarmCar.FillColor = AlarmDesc.FillColor
    AlarmDesc.FillColor = AlarmCent.FillColor
    AlarmLat.FillColor = AlarmCar.FillColor
    AlarmInsp.FillColor = AlarmCar.FillColor
'Geração de Mensagens de advertência

```

```

Mensagem.Caption = ""
If Estágio(2).Visible <> S_COD1 Then Mensagem.Caption = "Erro no Sensor Cod1"
If Estágio(3).Visible <> S_CENT Then Mensagem.Caption = "Erro no Sensor Central"
If Estágio(4).Visible <> S_CURVA1 Then Mensagem.Caption = "Erro no Sensor Curva1"
If Estágio(5).Visible <> S_LAT Then Mensagem.Caption = "Erro no Sensor Lateral"
If Estágio(7).Visible <> S_INSP Then Mensagem.Caption = "Erro no Sensor Inspeção"
If Estágio(8).Visible <> S_CURVA2 Then Mensagem.Caption = "Erro no Sensor Curva2"
If Estágio(9).Visible <> S_DESC Then Mensagem.Caption = "Erro no Sensor Descar"
If Image1.Visible = Elev_Cent Then Mensagem.Caption = "Erro no Sensor Elev_Cent"
If Image2.Visible = Elev_Lat Then Mensagem.Caption = "Erro no Sensor Elev_Lat"
If Image3.Visible = Elev_Insp Then Mensagem.Caption = "Erro no Sensor Elev_Insp"
SS_Cod1.Visible = S_COD1
SS_Cent.Visible = S_CENT
SS_Curva1.Visible = S_CURVA1
SS_Lat.Visible = S_LAT
SS_Insp.Visible = S_INSP
SS_Curva2.Visible = S_CURVA2
SS_Desc.Visible = S_DESC
SS_Elev_Cent.Visible = Elev_Cent
SS_Elev_Lat.Visible = Elev_Lat
SS_Elev_Insp.Visible = Elev_Insp
F_CB.Visible = Fazer_CB
F_Cent.Visible = Fazer_Cent
F_Lat.Visible = Fazer_Lat
F_Insp.Visible = Fazer_Insp
F_Desc.Visible = Fazer_Desc

```

'Lógica 1 - Carregamento

Select Case Logical

Case 1

'Atualizar Bando de Dados

'Procurar no banco de dados

Dados.Refresh

Dados.Recordset.MoveFirst

Dados.Recordset.FindFirst ("Est1=0")

aux1 = Dados.Recordset("Est1")

Erro = False

If aux1 <> 0 Then

Mensagem = "Não existe Est1 = 0" & Chr(10)

Mensagem = Mensagem & "Não era esperada base nesta etapa"

Logica1 = 0

```

    Mensagem.Caption = Mensagem
    Erro = True
Else
    Dados.Recordset.Edit
    Dados.Recordset("Est1") = Time
    Dados.Recordset.Update
    Logical = Logical + 1
    Estágio(0).Visible = True
End If
Case 2
    If Fazer_CB = False And S_COD1 = False Then
        Tempo_CB = Timer + 4
        Espera_CB.Visible = True
        CARREGA = True
        Estágio(0).Visible = False
        Estágio(1).Visible = True
        Logical = Logical + 1
    End If
Case 3
    aux1 = Tempo_CB - Timer
    Espera_CB.Caption = aux1
    If aux1 < 0 Then
        Logical = 1
        Espera_CB.Visible = False
        Estágio(1).Visible = False
        Estágio(11).Visible = True
        Fazer_CB = True
        Codigol.Enabled = True
        Codigol.SetFocus
    End If
Case Else
    Logical = 0
End Select

```

'Lógica 2 - Leitura do Primeiro Código de Barras

```
Select Case Logica2
```

```
Case 1
```

```
'Esperar por Leitura do Código de Barras
```

```
Case 2
```

```
'Esperar por peça no sensor 1 e esperar a peça que
```

```
'teve seu cód. lido sair do sensor
```

```
If S_COD1 Then
    Logica2 = Logica2 + 1
    CARREGA = False
    Estágio(11).Visible = False
    Estágio(2).Visible = True
End If
Case 3
'Atualizar o Banco de Dados
Call NovoEstágio(2)
If Erro Then
    Estágio(2).Visible = False
    Logica2 = 1
Else
    Dados.Recordset.Edit
    Dados.Recordset("CodBar1") = Barcod1
    Dados.Recordset.Update
    Logica2 = Logica2 + 1
End If
Case 4
If Fazer_Cent = False And S_CENT = False Then
    Tempo_Cent = Timer + 4
    Espera_Cent.Visible = True
    Fazer_CB = False
    Logica2 = Logica2 + 1
End If
Case 5
aux1 = Tempo_Cent - Timer
Espera_Cent.Caption = aux1
If aux1 < 0 Then
    Espera_Cent.Visible = False
    Fazer_Cent = True
    CENT = False
    Logica2 = Logica2 + 1
End If
Case 6
If S_COD1 = False Then
    Logica2 = Logica2 + 1
    Estágio(2).Visible = False
    Estágio(12).Visible = True
End If
Case Else
```

```
Logica2 = 1
```

```
End Select
```

```
'Lógica 3 - Mont./Desmont. Central
```

```
'Formulário: Central.frm
```

```
Select Case Logica3
```

```
Case 1
```

```
'Esperar por peça no sensor 2 e saída da base montada
```

```
  If S_CENT Then
```

```
    Logica3 = Logica3 + 1
```

```
    Estágio(12).Visible = False
```

```
    Estágio(3).Visible = True
```

```
    'Procurar no banco de dados
```

```
    Call NovoEstágio(3)
```

```
    If Erro Then
```

```
      Logica3 = 1
```

```
      Estágio(3).Visible = False
```

```
    Else
```

```
      Logica3 = Logica3 + 1
```

```
      'Leitura do Código de Montagem
```

```
      CodigoDeMontagem (CodMont)
```

```
      'Carregar as figuras
```

```
      Figura = "0" & Tb(1) & "0" & "f"
```

```
      Imagem1.Picture = ImageList1.ListImages(Figura).Picture
```

```
      Imagem1.Visible = False
```

```
      'Configurando os sinais de saída p/ posto Central
```

```
      If Tb(1) = 1 Then
```

```
        CENT = True
```

```
        CENT_12 = False
```

```
      ElseIf Tb(1) = 2 Then
```

```
        CENT = True
```

```
        CENT_12 = True
```

```
      Else
```

```
        CENT = False
```

```
      End If
```

```
      CENT_MD = Montag
```

```
      Central.Visible = True
```

```
      FigCentral = "Cubos" + Format(Tb(1))
```

```
      Central.Picture = ImageList2.ListImages(FigCentral).Picture
```

```
    End If
```

```
End If
```

Case 2

Logica3 = Logica3 + 1

Case 3

Logica3 = Logica3 + 1

Case 4

Logica3 = Logica3 + 1

Case 5

aux1 = (S_CENT = False And Elev_Cent = False) Or CENT = False

If aux1 Then

* Esperar por desativação do S_CENT e do Elev_Cent

Logica3 = Logica3 + 1

Estágio(3).Visible = False

Image1.Visible = True

Central.Visible = False

CENT = False

End If

Case 6

If Elev_Cent = True Then

* Esperar por desativação do Elev_Cent

Logica3 = Logica3 + 1

Estágio(3).Picture = Image1.Picture

Estágio(3).Visible = True

Image1.Visible = False

End If

Case 7

If Fazer_Lat = False And S_LAT = False Then

Tempo_Lat = Timer + 4

Espera_Lat.Visible = True

Fazer_Cent = False

LATERAL_E = False

LATERAL_D = False

Logica3 = Logica3 + 1

End If

Case 8

aux1 = Tempo_Lat - Timer

Espera_Lat.Caption = aux1

If aux1 < 0 Then

Espera_Lat.Visible = False

Logica3 = Logica3 + 1

Fazer_Lat = True

End If

Case 9

If S_CENT = False Then

'Final do processo de montagem central

Logica3 = Logica3 + 1

Figura = "0" & "0" & "0" & "f"

Estágio(3).Picture = ImageList1.ListImages(Figura).Picture

Estágio(3).Visible = False

Estágio(13).Picture = Image1.Picture

Estágio(13).Visible = True

End If

Case Else

Logica3 = 1

End Select

'Lógica 4 - Lógica intermediária 3-5 p/ separação de

'bases muito próximas

Select Case Logica4

Case 1

'Esperar por peça no sensor curval

If S_CURVA1 Then

Logica4 = Logica4 + 1

Estágio(13).Visible = False

Estágio(4).Visible = True

'Procurar no bando de dados

Call NovoEstágio(4)

If Erro Then

Logica4 = 1

Estágio(4).Visible = False

Else

'Leitura do Código de Montagem

CodigoDeMontagem (CodMont)

'Carregar as imagens

Figura = "0" & Tb(1) & "0" & "f"

Estágio(4).Picture = ImageList1.ListImages(Figura).Picture

End If

End If

Case 2

If S_CURVA1 = False Then

Logica4 = Logica4 + 1

Estágio(4).Visible = False

Estágio(14).Picture = Estágio(4).Picture

```

    Estágio(14).Visible = True
End If
Case Else
Logica4 = 1
End Select

```

```

'Lógica 5 - Mont./Desmont. Lateral

```

```

Select Case Logica5

```

```

Case 1

```

```

'Esperar por peça no sensor M/D lateral

```

```

If S_LAT Then

```

```

    Logica5 = Logica5 + 1

```

```

    Estágio(14).Visible = False

```

```

    Estágio(5).Visible = True

```

```

'Procurar no banco de dados

```

```

Call NovoEstágio(5)

```

```

Call NovoEstágio(6)

```

```

'Leitura do Código de Montagem

```

```

If Erro Then

```

```

    Logica5 = 1

```

```

    Estágio(5).Visible = False

```

```

Else

```

```

    CodigoDeMontagem (CodMont)

```

```

'Carregar as figuras

```

```

Figura = Tb(2) & Tb(1) & Tb(0) & "f"

```

```

Image2.Picture = ImageList1.ListImages(Figura).Picture

```

```

Image2.Visible = False

```

```

LateralE.Visible = True

```

```

LateralD.Visible = True

```

```

FigLatE = "Cubos" + Format(Tb(2))

```

```

FigLatD = "Cubos" + Format(Tb(0))

```

```

LateralE.Picture = ImageList2.ListImages(FigLatE).Picture

```

```

LateralD.Picture = ImageList2.ListImages(FigLatD).Picture

```

```

Figura = "0" & Tb(1) & "0" & "f"

```

```

Estágio(5).Picture = ImageList1.ListImages(Figura).Picture

```

```

'Configurando os sinais de saída da Lat. Esquerda

```

```

If Tb(2) = 1 Then

```

```

    LATERAL_E = True

```

```

    LATE_12 = False

```

```

ElseIf Tb(2) = 2 Then

```

```

    LATERAL_E = True

```

```

    LATE_12 = True
Else
    LATERAL_E = False
End If
LATE_MD = Montag
'Configurando os sinais de saída da Lat. Direita
If Tb(0) = 1 Then
    LATERAL_D = True
    LATD_12 = False
ElseIf Tb(0) = 2 Then
    LATERAL_D = True
    LATD_12 = True
Else
    LATERAL_D = False
End If
    LATD_MD = Montag
End If
End If
Case 2
    Logica5 = Logica5 + 1
Case 3
    'Esperar por desativação/ativação
    aux1 = S_LAT = False And Elev_Lat = False Or LATERAL_E = False And LATERAL_D = False
    If aux1 Then
        Logica5 = Logica5 + 1
        LateralD.Visible = False
        LateralE.Visible = False
        Image2.Visible = True
        Estágio(5).Visible = False
        LATERAL_D = False
        LATERAL_E = False
    End If
Case 4
    'Esperar por desativação/ativação
    If Elev_Lat Then
        Logica5 = Logica5 + 1
        Image2.Visible = False
        Estágio(5).Picture = Image2.Picture
        Estágio(5).Visible = True
    End If
Case 5

```

```

If Fazer_Insp = False And S_INSP = False Then
    Tempo_Insp = Timer + 4
    Espera_Insp.Visible = True
    Fazer_Lat = False
    Logica5 = Logica5 + 1
End If
Case 6
    aux1 = Tempo_Insp - Timer
    Espera_Insp.Caption = aux1
    If aux1 < 0 Then
        Espera_Insp.Visible = False
        Logica5 = Logica5 + 1
        Fazer_Insp = True
        INSPECAO = True
    End If
Case 7
    If S_LAT = False Then
        Logica5 = Logica5 + 1
        Estágio(15).Picture = Image2.Picture
        Estágio(15).Visible = True
        Estágio(5).Visible = False
    End If
Case Else
    Logica5 = 1
End Select

```

'Lógica 6 - Leitura do Segundo Código de Barras

"Select Case Logica6

"Case 1

'Esperar por peça no sensor

"If S_COD2 Then

"Logica6 = Logica6 + 2

'Codigo2.Enabled = True

'Codigo2.SetFocus

"Estágio(6).Visible = True

"Estágio(15).Visible = False

"Estágio(6).Picture = Estágio(15).Picture

"End If

"Case 2

'Esperar por Leitura do Código de Barras

"Case 3

```

'Atualizar o Banco de Dados
"Call NovoEstágio(6)
"If Erro Then
  "Estágio(6).Visible = False
  "Logica6 = 1
"Else
  "Dados.Recordset.Edit
'  Dados.Recordset("CodBar2") = Barcod2
  "Dados.Recordset("CodBar2") = 0
  "Dados.Recordset.Update
"End If
"Logica6 = Logica6 + 1
' Codigo2.Enabled = False
"Case 4
  "If S_COD2 = False Then
    "Logica6 = Logica6 + 1
    "Estágio(6).Visible = False
    "Estágio(16).Picture = Estágio(6).Picture
    "Estágio(16).Visible = True
  "End If
"Case Else
"Logica6 = 1
"End Select

```

'Lógica 7 - Leitura do Código de Inspeção

'Formulário: Inspecao.frm

Select Case Logica7

Case 1

'Esperar por peça no sensor

If S_INSP Then

'Procurar no banco de dados

Logica7 = Logica7 + 1

Estágio(7).Picture = Estágio(15).Picture

Estágio(15).Visible = False

'Estágio(16).Visible = False

Estágio(7).Visible = True

REFUGO = False

INSPECAO = False

End If

Case 2

'Procurar no banco de dados

```

Dados.Refresh
Dados.Recordset.MoveFirst
Dados.Recordset.FindFirst ("Est7=0")
aux1 = Dados.Recordset("Est7")
Erro = False
If aux1 <> 0 Then
    Mensagem = "Não existe " & Est & " = 0" & Chr(10)
    Mensagem = Mensagem & "Não era esperada base nesta etapa"
    Mensagem.Caption = Mensagem
    Erro = True
    Logica7 = 1
End If
aux1 = Dados.Recordset("Est5")
If aux1 = 0 Then
    Mensagem = "Etapa anterior, " & Estant & " =0"
    Mensagem = Mensagem & Chr(10) & "Base em posição indevida"
    Mensagem.Caption = Mensagem
    Erro = True
    Logica7 = 1
End If
If Not Erro Then
    'Leitura do Código de Montagem
    CodMont = Dados.Recordset("CodMont")
    CodigoDeMontagem (CodMont)
    'Armazena os valores trinários p/ posterior verificação
    CodInsp1 = Tb(0)
    CodInsp2 = Tb(1)
    CodInsp3 = Tb(2)
    'Carregar as figuras
    Figura = Tb(2) & Tb(1) & Tb(0) & "f"
    Estágio(7).Picture = ImageList1.ListImages(Figura).Picture
    Logica7 = Logica7 + 1
End If
Case 3
    If S_INSP = False And Elev_Insp = False Then
        Logica7 = Logica7 + 1
    End If
Case 4
    'Tirar a base da esteira
    Image3.Picture = Estágio(7).Picture
    Image3.Visible = True

```

```

Estágio(7).Visible = False
'Formulário de Leitura do Código de Inspeção
For i = 0 To 5
    B(i).Visible = True
    BL(i).Visible = True
    BLine(i).Visible = True
Next i
'Entrada dos códigos de inspeção
B(0) = -B11
B(1) = -B12
B(2) = -B21
B(3) = -B22
B(4) = -B31
B(5) = -B32
TbInsp1 = B(0) + 2 * B(1)
TbInsp2 = B(2) + 2 * B(3)
TbInsp3 = B(4) + 2 * B(5)
If TbInsp1 <> CodInsp1 Then
    Mensagem.Caption = "Lateral direita montada errada"
    ErroMontagem (0)
Else
    BL(0).BackColor = QBColor(15) 'White
End If
If TbInsp2 <> CodInsp2 Then
    Mensagem.Caption = "Centro montado errado"
    ErroMontagem (2)
Else
    BL(2).BackColor = QBColor(15) 'White
End If
If TbInsp3 <> CodInsp3 Then
    Mensagem.Caption = "Lateral esquerda montada errada"
    ErroMontagem (4)
Else
    BL(4).BackColor = QBColor(15) 'White
End If
If Elev_Insp Then
    Logica7 = Logica7 + 1
End If
Case 5
'Recolocar a base na esteira
Image3.Visible = False

```

```

Estágio(7).Visible = True
'Calcular CodInsp em função dos B(i)
'Entrada dos códigos de inspeção
Call NovoEstágio(7)
TbInsp1 = -B11 - 2 * B12
TbInsp2 = -B21 - 2 * B22
TbInsp3 = -B31 - 2 * B32
CodInsp = TbInsp3 * 3 ^ 2 + TbInsp2 * 3 + TbInsp1
If CodInsp > 26 Then CodInsp = 26
If Not Erro Then
  Dados.Recordset.Edit
  Dados.Recordset("CodInsp") = CodInsp
  Dados.Recordset.Update
  'Verifica se a peça é boa
  If CodMont <> CodInsp Then
    REFUGO = True
    Dados.Recordset.Edit
    Dados.Recordset("Montag") = False
    '
    Dados.Recordset("Est8") = #12:00:01 AM#
    '
    Dados.Recordset("Est9") = #12:00:01 AM#
    '
    Dados.Recordset("Est10") = #12:00:01 AM#
    Dados.Recordset.Update
    'Código de Montagem em base 3
    CodigoDeMontagem (CodInsp)
    'Carregar as figuras
    Figura = Tb(2) & Tb(1) & Tb(0) & "f"
    Estágio(7).Picture = ImageList1.ListImages(Figura).Picture
  End If
  'Formulário de Leitura do Código de Inspeção
  For i = 0 To 5
    B(i).Visible = False
    BL(i).Visible = False
    BLine(i).Visible = False
  Next i
End If
Logica7 = Logica7 + 1
Case 6
If Fazer_Desc = False And S_DESC = False Then
  Tempo_Desc = Timer + 4
  Espera_Desc.Visible = True
  Fazer_Insp = False

```

```

    Logica7 = Logica7 + 1
End If
Case 7
    aux1 = Tempo_Desc - Timer
    Espera_Desc.Caption = aux1
    If aux1 < 0 Then
        Espera_Desc.Visible = False
        Logica7 = Logica7 + 1
        Fazer_Desc = True
    End If
Case 8
    If S_INSP = False Then
        Estágio(7).Visible = False
    '    If Not REFUGO Then
        Estágio(17).Picture = Estágio(7).Picture
        Estágio(17).Visible = True
    '    End If
        Logica7 = Logica7 + 1
    End If
Case Else
    Logica7 = 1
End Select

```

'Lógica 8 - Lógica intermediária 7-9 p/ separação de
'bases muito próximas

```

Select Case Logica8
Case 1
'Esperar por peça no sensor curva2
    If S_CURVA2 Then
        Logica8 = Logica8 + 1
        Estágio(17).Visible = False
        Estágio(8).Visible = True
        'Procurar no banco de dados
        Call NovoEstágio(8)
        If Erro Then
            Logica8 = 1
            Estágio(8).Visible = False
        Else
            'Leitura do Código de Montagem
            CodigoDeMontagem (CodInsp)
            'Carregar as figuras

```

```

    Figura = Tb(0) & Tb(1) & Tb(2) & "f"
    Estágio(8).Picture = ImageList1.ListImages(Figura).Picture
    Dados.Refresh
    Dados.Recordset.MoveLast
End If
End If
Case 2
If S_CURVA2 = False Then
    Logica8 = Logica8 + 1
    Estágio(8).Visible = False
    Estágio(18).Picture = Estágio(8).Picture
    Estágio(18).Visible = True
End If
Case Else
Logica8 = 1
End Select

'Lógica 9 - Descarregamento
Select Case Logica9
Case 1
'Esperar por peça no sensor de descarga
If S_DESC Then
    S_DESC = False
    Logica9 = Logica9 + 1
    Estágio(18).Visible = False
    Estágio(9).Visible = True
    'Procurar no banco de dados
    Call NovoEstágio(9)
    If Erro Then
        Logica9 = 1
        Estágio(9).Visible = False
    Else
        'Leitura do Código de Montagem
        CodigoDeMontagem (CodInsp)
        'Carregar as imagens
        Figura = Tb(0) & Tb(1) & Tb(2) & "f"
        Estágio(9).Picture = ImageList1.ListImages(Figura).Picture
        DESCARREGA = True
    End If
End If
Case 2

```

```

If S_DESC = False Then
    Logica9 = Logica9 + 1
    Estágio(9).Visible = False
    Logica10 = 1
End If
Case Else
Logica9 = 1
End Select

'Lógica10 - Saída da base
Select Case Logica10
Case 1
    Logica10 = Logical10 + 1
    'Procurar no banco de dados
    Call NovoEstágio(10)
    Estágio(10).Visible = True
    Estágio(10).Picture = Estágio(9).Picture
    DESCARREGA = False
    Fazer_Desc = False
Case Else
    Logica10 = 0
End Select

'Tradução das Saídas
Bout(1) = CARREGA 'Car. da placa da base no sist. transf.
Bout(2) = LATERAL_D
Bout(3) = LATERAL_E
Bout(4) = CENT
Bout(5) = INSPECAO
Bout(6) = DESCARREGA
Bout(7) = REFUGO
Bout(8) = CENT_MD
Bout(9) = CENT_12
Bout(10) = LATD_MD
Bout(11) = LATD_12
Bout(12) = LATE_MD
Bout(13) = LATE_12
Bout(14) = Fazer_CB
Bout(15) = Fazer_Cent
Bout(16) = Fazer_Lat
Bout(17) = Fazer_Insp

```

```

    Bout(18) = Fazer_Desc
'Envio dos dados
Escrita
End Sub

```

```

Public Sub Leitura()
'Leitura do Teclado
    If Not Testar Then Exit Sub
    If nt > 26 Then nt = nt - 32
    If nt < 1 Or nt > 21 Then Exit Sub
    If Bin(nt) = -1 Then
        Bin(nt) = 0
    Else
        Bin(nt) = -1
    End If

```

```

'Reset do n. lido pelo teclado
nt = 0

```

```

'Leitura da Porta de Entrada
'If Not (Testar) Then
'    Dim InValue As Byte
'    InValue = PortIn(&H378)
'    For i = 0 To 7
'        If (InValue And (2 ^ i)) Then
'            Bin(i) = True
'        Else
'            Bin(i) = False
'        End If
'    Next i
'End If
End Sub

```

```

Public Sub Escrita()
'Calculando o valor decimal p/ envio porta2
'    Dim Soma As Byte
'    Soma = 0
'    For i = 0 To 7
'        Soma = -Bout(i + 1) * 2 ^ i + Soma
'    Next i
'Call PortOut(&H378, Soma)
'    Soma = 0
'    For i = 0 To 7

```

```
' Soma = -Bout(i + 9) * 2 ^ i + Soma
' Next i
'Call PortOut(&H37A, Soma)
End Sub
```

```
Public Sub NovoEstágio(Estágio)
```

```
'Procurar no banco de dados
```

```
Dados.Refresh
```

```
Dados.Recordset.MoveFirst
```

```
Est = "Est" & Format(Estágio)
```

```
Dados.Recordset.FindFirst (Est & "=0")
```

```
aux1 = Dados.Recordset(Est)
```

```
If aux1 <> 0 Then
```

```
    Mensagem = "Não existe " & Est & " = 0" & Chr(10)
```

```
    Mensagem = Mensagem & "Não era esperada base nesta etapa"
```

```
    Mensagem.Caption = Mensagem
```

```
    Erro = True
```

```
    Exit Sub
```

```
End If
```

```
Estant = "Est" & Format(Estágio - 1)
```

```
aux1 = Dados.Recordset(Estant)
```

```
If aux1 = 0 Then
```

```
    Mensagem = "Etapa anterior, " & Estant & " =0"
```

```
    Mensagem = Mensagem & Chr(10) & "Base em posição indevida"
```

```
    Mensagem.Caption = Mensagem
```

```
    Erro = True
```

```
    Exit Sub
```

```
End If
```

```
Erro = False
```

```
CodMont = Dados.Recordset("CodMont")
```

```
CodBar1 = Dados.Recordset("CodBar1")
```

```
CodBar2 = Dados.Recordset("CodBar2")
```

```
CodInsp = Dados.Recordset("CodInsp")
```

```
Montag = Dados.Recordset("Montag")
```

```
Dados.Recordset.Edit
```

```
Dados.Recordset(Est) = Time
```

```
Dados.Recordset.Update
```

```
End Sub
```

```
Public SubCodigoDeMontagem(CodMont)
```

```
'Cálculo do Código de Montagem
```

```
' Transformacao decimal -> trinário
aux1 = CodMont
For i = 2 To 0 Step -1
    Tb(i) = Fix(aux1 / (3 ^ i))
    aux1 = aux1 - Tb(i) * (3 ^ i)
Next i
End Sub

Public Function ErroMontagem(Bi)
    If BL(Bi).BackColor = QBColor(12) Then 'Light RED
        BL(Bi).BackColor = QBColor(10) 'Light Green
    Else
        BL(Bi).BackColor = QBColor(12) 'Light RED
    End If
End Function
```

Anexo V - Interfaces de Comunicação

Interface Paralela

A interface paralela normalmente é utilizada para conexão da impressora ou outros periféricos, como “scanners”, câmara de vídeo, etc. Mas ela também pode ser utilizada para aquisição de dados digitais, como será mostrado neste anexo. Ela é composta de três portas, a saber:

- 378 – Utilizada normalmente para envio de dados. No caso da utilização desta porta para leitura de dados, o dispositivo externo deverá enviar um sinal alto (em contraposição ao modo normal de utilização da porta paralela, que é o envio do terra - nunca enviar um sinal baixo para esta porta); neste caso, os valores lidos por esta porta serão combinados, por uma operação OU, com os últimos valores presentes na porta.
- 379 – Utilizada apenas para leitura de dados, devendo ser obedecido, pelo dispositivo externo, o padrão de envio do terra.
- 37A – Utilizada tanto para leitura como para envio de dados, devendo ser obedecido, pelo dispositivo externo, o padrão de envio do terra.

A ligação da porta paralela com os dispositivos externos é feita através de um conector de 25 pinos (DB25), cuja especificação de cada pino é mostrada na Tabela 8.4 e na Figura 8.5.

Para a construção de um cabo de comunicação paralela do tipo Lap-Link ou Inter-Link, deve-se utilizar os pinos de 2, 3, 4, 5, 6 e 25 para envio de dados e os conectar respectivamente com os pinos 15, 13, 12, 10, 11 e 25 para recebimento. Utilize conectores DB25 macho em ambos os lados. Onze fios são requeridos para a transmissão de dados. No envio de dados, deve-se chavear para um sinal baixo (GND), pois o estado normal é alto, como pode ser observado

quando o cabo da impressora está desconectado do computador, e os pinos 1, 10, 11, 12, 13 e 15 estão no estado alto.

Tabela 8.4: Especificação dos Pinos da Porta Paralela

Micro DB25	Direção	Impressora Centronics	Nome	Porta – Bit de Dados	Designação (vide Figura 8.5)
1	↔	1	/Strobe	37A – B0 ¹	$\overline{C0}$
2	⇒ *	2	Data0	378 – B0	D0
3	⇒ *	3	Data1	378 – B1	D1
4	⇒ *	4	Data2	378 – B2	D2
5	⇒ *	5	Data3	378 – B3	D3
6	⇒ *	6	Data4	378 – B4	D4
7	⇒	7	Data5	378 – B5	D5
8	⇒	8	Data6	378 – B6	D6
9	⇒	9	Data7	378 – B7	D7
10	⇐ *	10	/Acknowledge	379 – B6	S6
11	⇐ *	11	Busy	379 – B7 ¹	$\overline{S7}$
12	⇐ *	12	Paper Out	379 – B5	S5
13	⇐ *	13	Select	379 – B4	S4
14	↔	14	/Autofeed	37A – B1 ¹	$\overline{C1}$
		15	n/c		
		16	0V		
		17	Chassis GND		
		18	+5V DC		
18		19	GND		

18		20	GND		
18		21	GND		
18		22	GND		
18		23	GND		
18		24	GND		
18		25	GND		
18		26	GND		
18		27	GND		
18		28	GND		
18		29	GND		
18		30	/GND - Reset		
16	↔	31	/Reset	37A – B2	C2
15	← *	32	/Fault	379 – B3	S3
		33	0V		
		34	n/c		
		35	+5V		
17	↔	36	/Select in	37A – B3 ¹	$\overline{C3}$

* Pino utilizado pelo cabo de comunicação de dados (tipo interlink)

¹ Pino com voltagem invertida

Para facilitar a ligação dos cabos no conector DB25, é mostrado na Figura 8.5 a designação dos pinos agrupados pelas portas, ou seja, pinos designados com D são pertencentes a porta 378, pinos designados com C são pertencentes a porta 37A e pinos designados com S são pertencentes a porta 379.

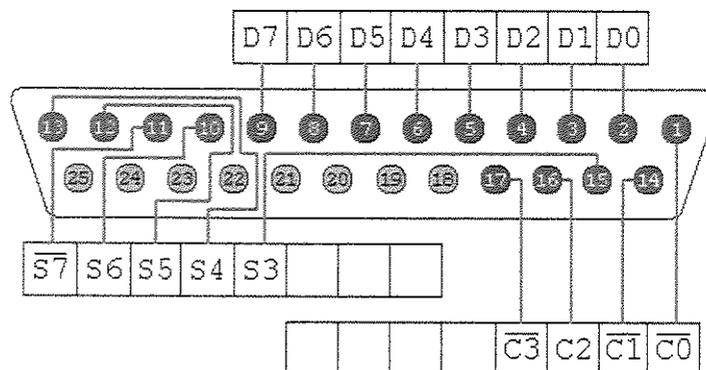


Figura 8.5: Especificação do conector Fêmea DB25

Interface de Comunicação Koyo

O CLP da Koyo possui uma interface de comunicação com o microcomputador, que permite a aquisição de dados por aplicativos Windows®. Ele utiliza a porta serial do microcomputador para o envio e o recebimento de dados. Para a comunicação com este CLP, foi utilizado o software de comunicação DirectSoft – DDE Server, fornecido pela Koyo Electronics Industries Co. Ltd. Apesar do software ser muito fácil de se utilizar, tivemos problemas para ajustar os parâmetros de comunicação, por falta de literatura sobre o assunto. Para utilização deste software necessitou-se configurar a porta serial e o protocolo de comunicação como mostrado nas Figura 8.6 e Figura 8.7.

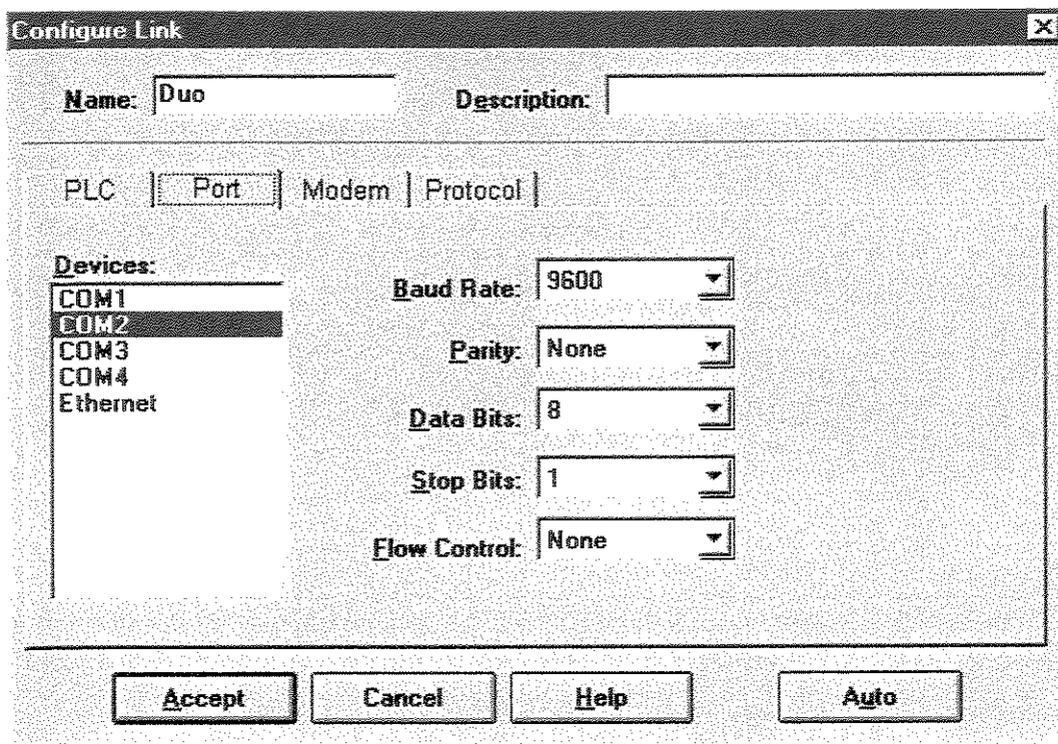


Figura 8.6: Link Serial com CLP Koyo

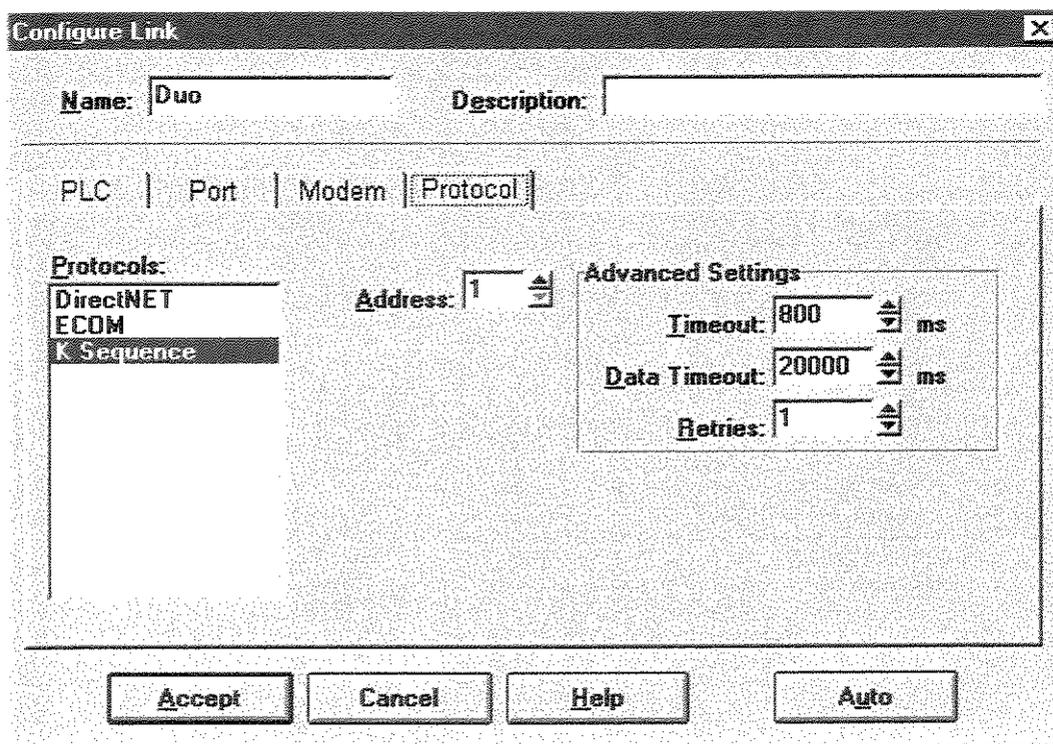


Figura 8.7: Protocolo de Comunicação Koyo

Esta interface de comunicação utiliza a porta serial do microcomputador, para fazer uma troca dinâmica de dados, em tempo real, entre os dados contidos no CLP e o microcomputador. Dessa forma, para enviar ou receber dados do CLP, deve-se executar um aplicativo que trabalhe com o protocolo DDE.

A configuração do Link de comunicação entre o microcomputador e o CLP pode ser feito utilizando-se três protocolos diferentes. O protocolo DirectNET e ECOM são mais rápidos do que o K Sequence, mas possuem certas limitações. Algumas funções não podem ser executadas, e tampouco pode-se escrever um bit isolado. Já o K Sequence não possui tantas limitações, mas só se pode utilizá-lo em uma porta do CLP por vez. Os teste realizados utilizando-se o DirectNET apresentaram problemas de envio de dados, por isto optou-se por utilizar o protocolo K Sequence.

Anexo VI – Rotinas de Apoio

Programa para Aquisição de Dados da Interface Paralela

A seguir é apresentado um programa para teste da comunicação, utilizando-se a interface paralela. Este programa é composto por dois formulários (um de entrada de dados e outro de saída) e um módulo básico, apresentados a seguir:

Formulário de Entrada de dados:

Option Explicit

```
Private Sub Form_Load()
```

```
    Dim i As Integer
```

```
    For i = 0 To 7
```

```
        Pin(i).FillColor = QBColor(10)
```

```
    Next i
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    Dim InValue As Byte
```

```
    Dim aux1 As Integer
```

```
    Dim i As Integer
```

```
    InValue = PortIn(&H379)
```

```
    Label1.Caption = InValue
```

```
    For i = 0 To 7
```

```
        If (InValue And (2 ^ i)) Then 'Test to see if Led is HIGH
```

```
            Pin(i).FillColor = QBColor(12) 'RED
```

```
        Else
```

```
            Pin(i).FillColor = QBColor(10) 'GREEN
```

```
        End If
```

```
    Next i
```

```
End Sub
```

Formulário de saída de dados:

```
Private Sub Command1_Click()
    End
End Sub
```

```
Private Sub Form_Load()
    Frmppin.Visible = True
End Sub
```

```
Private Sub Timer1_Timer()
    Dim i As Integer
    Dim Value As Byte
    Value = 0
    For i = 0 To 7
        If Chkbit(i).Value = 1 Then
            Value = Value + 2 ^ i
        End If
    Next i
    Call PortOut(&H378, Value)
End Sub
```

Módulo Básico:

```
Public Declare Sub PortOut Lib "io.dll" (ByVal Port As Integer, ByVal Value As Byte)
Public Declare Function PortIn Lib "io.dll" (ByVal Port As Integer) As Byte
```

A seguir é apresentado o “layout” dos dois formulários:

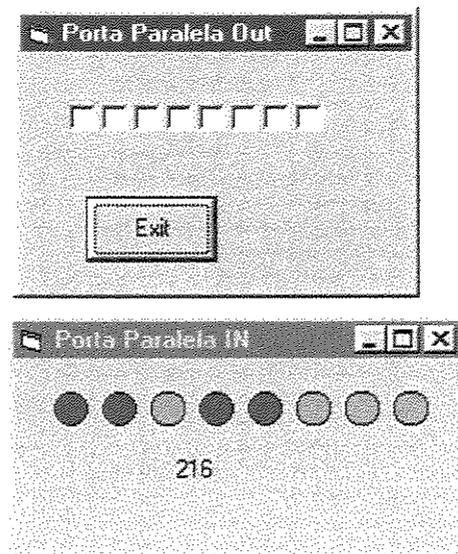


Figura 8.8: Aquisição de Dados da Porta Paralela

Aquisição de Dados do CLP da Koyo

Para teste da comunicação com o CLP da Koyo, desenvolveu-se um programa composto por apenas um formulário, que faz o envio e o recebimento de dados. O “layout” do formulário é mostrado na figura abaixo. Cada campo em branco do formulário, está ligado a uma sentença DDE. Por exemplo, para ler o valor da variável Y0 através do link Kseq, deve-se definir:

1. A propriedade LinkTopic para "DSDDE|Kseq";
2. A propriedade LinkItem para "Y0";
3. A propriedade LinkMode para 1 - Automatic.

Para o envio de dados procede-se de mesmo modo, mas os dados só serão enviados após o comando LinkPoke.

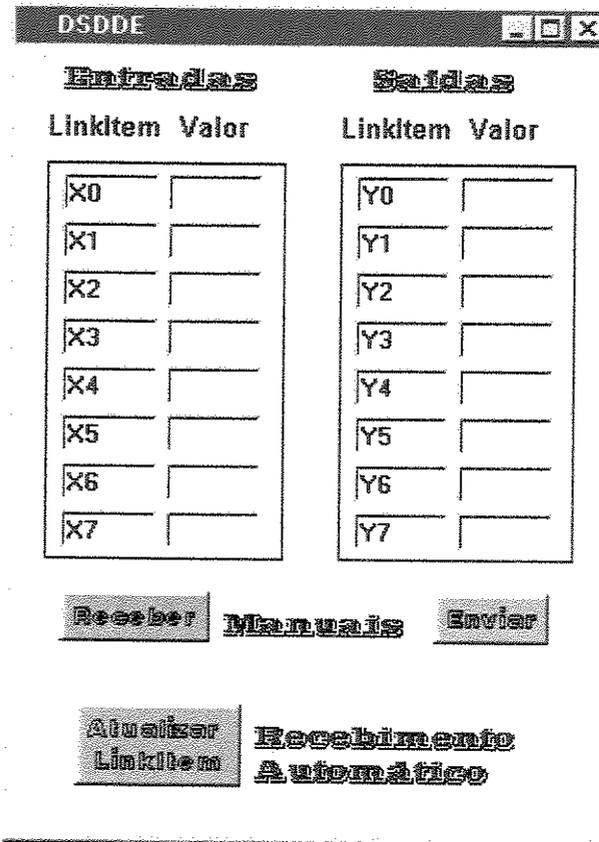


Figura 8.9: Aquisição de Dados do CLP da Koyo

Programa Principal para Troca de Dados com CLP Koyo:

```

Private Sub Automático_Click()
    For i = 0 To 7
        If Entrada(i).Text = "" Then
            X(i).LinkMode = 0
            X(i).Caption = ""
        Else
            X(i).LinkItem = Entrada(i).Text
            X(i).LinkMode = 1
        End If
    Next i
End Sub

```

```
Private Sub Enviar_Click()  
  For i = 0 To 7  
    If saída(i).Text = "" Or Y(i).Text = "" Then  
      Y(i).LinkMode = 0  
      Y(i).Text = ""  
    Else  
      Y(i).LinkItem = saída(i).Text  
      Y(i).LinkMode = 2  
      Y(i).LinkPoke  
    End If  
  Next i  
End Sub
```

```
Private Sub Receber_Click()  
  For i = 0 To 7  
    If Entrada(i).Text = "" Then  
      X(i).LinkMode = 0  
      X(i).Caption = ""  
    Else  
      X(i).LinkItem = Entrada(i).Text  
      X(i).LinkMode = 2  
      X(i).LinkRequest  
    End If  
  Next i  
End Sub
```

Anexo VII – Programa do Transporte de Peças em Diagrama de Relés

A seguir é apresentado o primeiro programa realizado no início deste trabalho. Ele foi desenvolvido com base em Linguagem de Relés, para ser multitarefa, realizando três tarefas aos mesmo tempo:

1. 'Lógica 1.1 - Desligar esteira 1;
2. 'Lógica 1.2 - Fornecer material para montagem;
3. 'Lógica 2. Intertravamento com a CLP.

A parte mais importante deste programa é a rotina Private Sub Timer1_Timer(), que realiza um “scanning” contínuo do sistema. É esta rotina que controla todo o funcionamento do sistema.

Dim Bin(16)

Dim Bout(16)

Dim nt

Dim Tempo1

Dim Tempo2

Dim Fim_Montagem

Dim Libera

Private Sub Escrita()

 IconBout(1).Visible = Bout(1)

 IconBout(3).Visible = Bout(3)

 IconBout(4).Visible = Bout(4)

 IconBout(5).Visible = Bout(5)

 IconBout(6).Visible = Bout(6)

 Label(11).Visible = Bout(1)

 Label(13).Visible = Bout(3)

 Label(14).Visible = Bout(4)

 Label(15).Visible = Bout(5)

 Label(16).Visible = Bout(6)

End Sub

```
Private Sub Form_KeyPress(KeyAscii As Integer)
```

```
    nt = KeyAscii - 48
```

```
End Sub
```

```
Private Sub IconBin_KeyPress(Index As Integer, KeyAscii As Integer)
```

```
    nt = KeyAscii - 48
```

```
End Sub
```

```
Private Sub IconBout_KeyPress(Index As Integer, KeyAscii As Integer)
```

```
    nt = KeyAscii - 48
```

```
End Sub
```

```
Private Sub Leitura()
```

```
    If Bin(nt) = True Then
```

```
        Bin(nt) = False
```

```
    Else
```

```
        Bin(nt) = True
```

```
    End If
```

```
    For i = 1 To 8
```

```
        IconBin(i).Visible = Bin(i)
```

```
        Label(i).Visible = Bin(i)
```

```
    Next i
```

```
    nt = 0
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    Leitura
```

```
    'Tradução de entradas
```

```
    Magnetico1 = Bin(1)
```

```
    Magnetico2 = Bin(2)
```

```
    SER = Bin(3) 'Sensor Elevador Recuado
```

```
    SEA = Bin(4) 'Sensor Elevador Atuado
```

```
    SGF = Bin(5) 'Sensor Gaveta Fechada
```

```
    SGA = Bin(6) 'Sensor Gaveta Aberta
```

```
    Magnetico3 = Bin(7)
```

```
    CLPIn = Bin(8) 'CLP pedindo descarga
```

```
    'Tradução de saída
```

```
    Motor1 = Bout(1)
```

```
    Motor2 = Bout(2)
```

```

Stop1 = Bout(3)
Elevador = Bout(4)
Stop2 = Bout(5)
CLPout = Bout(6)

```

'Lógica 1.1 - Desligar esteira 1

```

If Magnetico1 = -1 And Magnetico2 = -1 Then
    Motor1 = 0
Else
    Motor1 = -1
End If

```

'Lógica 1.2 - Fornecer material para montagem

```

Final = Not CLPin And Not CLPout
If Final And Magnetico2 Then 'acabou a montagem e tem peça na espera
    Stop1 = 0
    Tempol = Timer
End If
If (Tempol + 3 < Timer) Then
    Stop1 = -1
End If

```

'Lógica 2. Intertravamento com a CLP

'Inicio da Montagem

```

If Fim_Montagem And Magnetico3 Then 'acabou montagem e tem peça em posição de carga
    Elevador = True
    CLPout = True 'avisar CLP p/ inicio de montagem
End If
'Passa o processo p/ CLP
If SEA And SGA And Not CLPin Then 'SEA+SGA+CLP em carga
    Fim_Montagem = False
    Elevador = False
End If
'Espera CLP acabar montagem
If SER And SGA And CLPin Then 'SER+SGA+CLP pedindo descarga
    Elevador = True
End If
'Baixar a peça na esteira
If SEA And SGF And CLPin Then 'SEA+SGF+CLP pedindo descarga
    Elevador = False
End If

```

```
'Final da montagem
If Magnetico3 And Not Fim_Montagem Then 'Descarga
    Tempo2 = Timer
    Stop2 = False
    Libera = False
End If
'Temporização - Ao final do processo
If (Tempo2 + 3 < Timer) And Final And Not Libera Then
    Stop2 = True
    Fim_Montagem = True
    Libera = True
End If
'Desativar CLPout
If SER And SGF And CLPin Then 'No meio do processo
    CLPout = False
End If

'Tradução de saída
Bout(1) = Motor1
Bout(2) = Motor2
Bout(3) = Stop1
Bout(4) = Elevador
Bout(5) = Stop2
Bout(6) = CLPout
Escrita
End Sub
```

Anexo VIII – Revisão de Portas Lógicas

Tabela 8.5: Portas Lógicas

Porta Lógica	AND (E) $S = a \cdot b$			OR (OU) $S = a + b$			Exclus. OR (OU exclu.) $S = a \oplus b$			NAND (NÃO E) $S = \overline{a \cdot b}$			NOR (NÃO OU) $S = \overline{a + b}$		
Tabela Verdade	a	b	S	a	B	S	a	b	S	a	b	S	a	b	S
	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
	0	1	0	0	1	1	0	1	1	0	1	1	0	1	0
	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0
	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0
Equivalente Elétrico															
Equivalente em linguagem de relés															