

**MAURÍCIO KAZUO TANIGUTI
WAGNER DE OLIVEIRA CARVALHO**

*note final
7,0 (sete)*

15m

CLP DE BAIXO CUSTO PARA SISTEMAS DE AUTOMAÇÃO DE
PEQUENO PORTE

Projeto de Formatura
apresentado à Escola
Politécnica da Universidade
de São Paulo para conclusão
do curso de Engenharia
Mecatrônica

São Paulo
2004

DEDALUS - Acervo - EPMN



31600011108

Carvalho, Wagner de Oliveira

CLP de baixo custo para sistemas de automação de pequeno porte, por W.O. Carvalho, M.K. Taniguti -- São Paulo, 2004. 120 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1. Controle digital 2. Automação 3. Controladores Programáveis I. Taniguti, Maurício Kazuo II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III. t.

AGRADECIMENTOS

Antes de tudo, agradecemos à Deus pelas nossas conquistas. Agradecemos as nossas famílias pelo apoio total e incondicional aos nossos objetivos. Aos nossos amigos e professores do curso.

Eu, Wagner Carvalho, agradeço em especial ao amigo Rogério Vazquez pelas horas de estudo juntos principalmente nos últimos meses do curso. Agradeço também ao amigo Maurício K. Taniguti, meu parceiro de trabalho de formatura, que me esperou para fazermos esse trabalho e me surpreendeu pela força de vontade em terminar o projeto de uma forma que pudéssemos ficar orgulhosos. Agradeço também a inestimáveis amigos que deixei na Polônia, Michal, Dr. Grzegorz, Zymon, Prof. Swidel que me ajudaram no estágio e na elaboração do hardware deste CLP.

Eu, Maurício Kazuo Taniguti, agradeço o amigo, meu parceiro de trabalho de formatura, que me suportou nas horas difíceis. Agradeço aos meus amigos do Piratiniga, do Mortos Vivos e da Accenture pela força.

Agradecemos também ao Prof. Marcos R. Pereira Barretto, pela paciência e pelos conselhos sobre como fazer um CLP de verdade.

RESUMO

Os controladores lógico programáveis (CLP) são dispositivos eletrônicos aplicados no campo da automação para executar o controle de um sistema ou máquina as quais se deseja a automatização dos processos. Normalmente constituído de um microprocessador, memórias e uma eletrônica que permita a sua interligação com os outros componentes do sistema, esse dispositivo se tornou essencial para qualquer linha produtiva automatizada atualmente (Carvalho, 2002).

A meta do trabalho foi de desenvolver um dispositivo que fosse compatível com os CLPs existentes no mercado, tanto em termos dimensionais, como em recursos para o usuário final, buscando também um custo *coerente com produtos similares* que torne o produto viável.

O microCLP, como também é chamado, é específico para aplicações de pequeno porte onde um dos fatores ^a mais crítico para se decidir pela utilização de CLPs é o custo, já que ele precisa oferecer uma relação custo/benefício melhor do que outras alternativas. O microCLP, nesse caso, oferece recursos semelhantes aos CLPs de grande porte, mas em quantidade menor.

O resultado foi um microCLP que conseguiu agregar em uma mesma ^{discreta:} estrutura interfaces de I/O, comunicação, memória e interface de usuário, o que permite que ele possa ser aplicado em sistemas de pequeno porte sem a necessidade de ~~mais~~ dispositivos. *adiciona de condicionamento de sinais.*

ABSTRACT

This work deals with the subject Programmable Logic Controller (PLC) and micro-automation and the main goal is to develop a compact controller that works like others microPLCs available on the market.

The project involves hardware and software. The hardware ~~is~~ compatible with the characteristics of industrial automation and its devices. The software ~~subject~~ includes the firmware that works inside the microPLC and the interface used for elaborate the user's program that ~~describes~~ the logic of the systems ~~and is executed~~ by the controller.

It tries to follow the international rule described by the IEC61131 norm. This norm describes languages used with PLCs and the main characteristics, like hardware, components and the logic of control processing of this kind of device. The norm is divided in seven items, among these items, the one about the languages (IEC61131-3) is the most important and the most used. It describes five kinds of languages, two textual and three graphic, and describes also another view to the logic of control processing.

Among these languages, the project uses two of them, the Ladder Diagram and the Instruction List. Latter this document explains how those languages are used on the interface to describe the systems to be controlled. The Ladder Diagram is the language applied by the user and the IL is the result of the compilation of user's program and is the language that the controller understands.

This document includes explanation about the hardware design for all the components, a little introduction about the norm IEC, examples of uses of this microPLC and the relation between the Ladder Diagram and Instruction List.

SUMÁRIO

1. INTRODUÇÃO.....	7
1.1. MOTIVAÇÃO E JUSTIFICATIVA	8
1.2. OBJETIVO	9
1.3. ORGANIZAÇÃO DO TEXTO	9
2. FUNDAMENTOS	10
2.1 VISÃO GERAL DO MICROCLP	10
2.2 APLICAÇÕES DO MICROCLP	11
2.3 NORMAS TÉCNICAS EXISTENTES	13
2.4 LINGUAGENS DE PROGRAMAÇÃO DA IEC61131-3	15
2.4.1 <i>Diagrama de Relés (LD)</i>	16
2.4.2 <i>Instruction List (IL)</i>	18
2.5 ARQUITETURA BÁSICA DE HARDWARE	21
2.6 FUNCIONAMENTO DO CLP	24
3. ESPECIFICAÇÃO TÉCNICA DO MICROCLP	25
3.1. HARDWARE	25
3.1.1 <i>Componentes do MICROCLP</i>	27
3.2 LINGUAGENS DE PROGRAMAÇÃO	34
3.2.1 <i>LD para o microCLP</i>	34
3.2.2 <i>IL para o microCLP</i>	40
3.3 EXECUÇÃO DO PROGRAMA DE CONTROLE.....	41
3.4. FIRMWARE DO MICROCLP	42
3.5. INTERFACE DE PROGRAMAÇÃO.....	44
4. IMPLEMENTAÇÃO DO MICROCLP.....	46
4.1 ESTRUTURA FÍSICA.....	46
4.2. EDIÇÃO DE PROGRAMAS DE CONTROLE	49
4.3. TRATAMENTO DOS DADOS DO PROGRAMA DE CONTROLE.....	59
4.4. ESTRUTURA DO PROGRAMA DE CONTROLE.....	61
4.5. COMUNICAÇÃO COM O MICROCLP	66
5. TESTES DO PROTÓTIPO DO MICROCLP	70
5.1. TESTES DE HARDWARE	70
5.2. TESTE DA INTERFACE DE PROGRAMAÇÃO E DO FIRMWARE	71
6. CONCLUSÕES.....	76
7. REFERÊNCIAS.....	78
ANEXO A. MANUAL DO MICROCLP	
ANEXO B. MANUAL DA INTERFACE	

LISTA DE FIGURAS

Figura 2.1. LD com a lógica por contatos e bobinas	17
Figura 2.2. LD com bloco de funções.....	18
Figura 2.3. Diagrama da arquitetura de hardware envolvendo CLP	21
Figura 3.1. Diagrama com os componentes do microCLP	27
Figura 3.2. Caixa do modelo H53 da ITALTRONIC em seu formato original	28
Figura 3.3. Circuito do "driver" de potência para a saída digital	30
Figura 3.4. Circuito da entrada digital do microCLP	30
Figura 3.5. Circuito para sensores com sinal em voltagem de 0 a 10 VDC	31
Figura 3.6. Circuito para sensores com sinal em corrente de 4 a 20 mA	31
Figura 3.7. Circuito equivalente a interface de usuário	32
Figura 3.8. Circuito equivalente ao drive da comunicação serial	33
Figura 3.9. Esquema com as propriedades do bloco temporizador	35
Figura 3.10. Aplicação do bloco temporizador	36
Figura 3.11. Esquema com os parâmetros do bloco contador.....	37
Figura 3.12. Aplicação do bloco contador	37
Figura 3.13. Esquema com as propriedades do bloco entrada analógica	39
Figura 3.14. Aplicação do bloco entrada analógica.....	39
Figura 3.15. "loop" de execução do programa de controle.....	41
Figura 3.16. Funcionamento do firmware do microCLP	43
Figura 3.17. Tela de trabalho da interface de programação	44
Figura 3.18. Tela de trabalho da interface de programação com um exemplo	45
Figura 4.1. Foto da aparência externa do microCLP	46
Figura 4.2. Detalhe das laterais do microCLP	47
Figura 4.3. Foto com as duas placas eletrônicas do microCLP.....	47
Figura 4.4. Detalhe dos componentes da placa "driver"	48
Figura 4.5. Tela de trabalho da interface de programação.....	49
Figura 4.6. Comandos da interface de programação	50
Figura 4.7. Comando "novo programa"	51
Figura 4.8. Contato Normalmente Aberto	51
Figura 4.9. Contato Normalmente Fechado	52
Figura 4.10. Início da linha OR	52
Figura 4.11. Fechamento da linha OR	52
Figura 4.12. Linha apagada.....	53
Figura 4.13. Bloco contador.....	53
Figura 4.14. Representação das configurações do bloco contador.....	54
Figura 4.15. Bloco temporizador.....	54
Figura 4.16. Representação das configurações do bloco temporizador.....	54
Figura 4.17. Bloco entrada analógica	55
Figura 4.18. Representação das configurações do bloco entrada analógica	55
Figura 4.19. Exemplo de linha de instruções	56
Figura 4.20. Exemplo de uma linha adicional OR	56
Figura 4.21. Indicação da posição e linha em edição	57
Figura 4.22. Caixa de seleção para opção de nome das variáveis	57
Figura 4.23. Exemplo de linha inválida.....	58
Figura 4.24. Tabela gerada com base nas entradas do LD editado	59
Figura 4.25. Resultado da compilação do LD	60
Figura 4.27. Configuração equivalente a Tabela 4.9	65

Figura 4.28. Tela principal do programa "microCLP.exe"	66
Figura 4.29. Indicação da abertura da porta COM	67
Figura 4.30. Programa confirmando a comunicação com o microCLP	67
Figura 4.31. Mensagem de programa carregado no microCLP	68
Figura 5.1. Foto da placa de testes dos recursos do microCLP.....	70
Figura 5.2. Exemplo de um programa de controle em LD	71

LISTA DE TABELAS

Tabela 2.1. Categorias de CLP	10
Tabela 2.2. IL da equivalente ao diagrama da Figura 2.1	19
Tabela 3.1. Recursos do PIC18F452	26
Tabela 3.2. Variáveis que podem ser utilizadas na programação	34
Tabela 3.3. Resumo dos parâmetros do bloco temporizador	36
Tabela 3.4. Resumo dos parâmetros do bloco contador	38
Tabela 3.5. Resumo das configurações do bloco entrada analógica	40
Tabela 3.6. Instruções e códigos do IL para microCLP.....	40
Tabela 4.1. Resumo dos parâmetros do bloco contador	54
Tabela 4.2. Resumo das configurações do bloco temporizador	55
Tabela 4.3. Resumo das configurações do bloco entrada analógica.....	55
Tabela 4.4. Códigos utilizados na "Tabela de Entradas"	60
Tabela 4.5. Lista de variáveis presentes na programação	61
Tabela 4.6. Lista de códigos para os transformadores.....	62
Tabela 4.7. Lista de códigos equivalentes para instruções	63
Tabela 4.8. Lista de códigos para as variáveis do sistema.....	64
Tabela 4.9. Cabeçalho do programa de controle	64
Tabela 4.10. Exemplos de IL convertido para os códigos do microCLP	65
Tabela 4.11. Releção entre os códigos do primeiro byte e as funções do microCLP.....	69
Tabela 4.12. Estrutura da mensagem de programação dos códigos no microCLP	69
Tabela 5.1. Programa em IL do exemplo da Figura 5.1	72
Tabela 5.2. Lista de bytes equivalente ao programa exemplo da Tabela 5.1.....	74

LISTA DE EQUAÇÕES

EQUAÇÃO 2.1.....	19
EQUAÇÃO 5.1.....	73
EQUAÇÃO 5.2.....	73
EQUAÇÃO 5.3.....	73
EQUAÇÃO 5.4.....	73
EQUAÇÃO 5.5.....	73
EQUAÇÃO 5.6.....	73

LISTA DE ABREVIATURAS

CLP	Controlador Lógico Programável
IEC	International Electrotechnical Commission
SED	Sistema a Eventos Discretos
I/O	input/output
IL	Instruction List
SFC	Sequential Flow Chart
LD	Ladder Diagram
DC	corrente contínua
AC	corrente alternada
V	Volts
mA	mili Ampères

1. INTRODUÇÃO

Desde a década de 70, os CLP (Controladores Lógico Programáveis), têm substituído os dispositivos eletromecânicos e/ou pneumáticos na automação industrial (Hughes, 1989). O mercado de aplicações, desde então, tem crescido e os CLP passaram a ser aplicados em outras áreas da automação como a predial, residencial, controle de plantas industriais químicas ou mesmo controle de processos laboratoriais (Carvalho, 2002).

Atualmente, o CLP não só executa a lógica dos relés, como pode executar algoritmos complexos de controle de sistema a eventos discretos (SED) mais modernos como MFG por exemplo. Mas além dos algoritmos de controle o CLP oferece recursos diversos para interfaces digitais e analógico, comunicação de tempo real, comunicação de alto nível, monitoramento de processos e integração com outros níveis do sistema produtivo (Carvalho, 2002).

Para as outras áreas ou para aplicações de controle distribuído ou com pequena quantidade de processos, nem todas as características citadas são utilizadas. São aplicações onde se prioriza o baixo custo da solução de automação. São aplicações que necessitam da robustez de um CLP, mas com uma gama menor de recursos.

Esse é o mercado chamado de micro-automatização, que atualmente tem se expandido, principalmente na área de automação predial e residencial. Para esse mercado, os desenvolvedores de sistemas de controle requerem dispositivos de tamanho reduzido e com interfaces compactas para se adequarem aos sistemas de pequeno porte (Marchesan, 2004).

Esses controladores aplicados na micro-automatização são chamados de microCLP (Marchesan, 2004), que trazem para os ^{sistemas} projetos de pequeno porte as técnicas de processamento de controle já consagradas nos processos industriais.

Tanto para os CLP de porte maior quanto para os microCLP, as características dos dispositivos são baseadas em uma norma internacional (IEC61311), que determina características funcionais, recursos, método de processamento de controle e linguagens de programação.

1.1. Motivação e justificativa

A cada ano surgem ^{dispositivos} modelos mais versáteis e de maior capacidade de processamento. Sistemas de grande porte, incorporam processadores de até 400Mhz, que reduzem o tempo de processamento do programa de controle, isso permite que sejam ^{Incluídos um número razoável de} incorporados mais ^(Entradas e Saídas) entradas e saídas no CLP, tanto digitais quanto analógicas, chegando a mais de 4000 pontos de I/O. Atualmente, os módulos de comunicação disponibilizam a conexão com redes, o que possibilita a interação em tempo real com outros CLP da planta industrial. Para os níveis de supervisão e gerenciamento, os CLP disponibilizam comunicações do tipo [?] ethernet por cabeamento ou radio-freqüência que permite o tratamento de um grande volume de dados da produção, além de um monitoramento remoto e detalhado do processo. ^(ref.?)

Além desta área de alto desempenho, onde se busca o hardware mais rápido e com o maior número de recursos, a área de micro-automação envolve sistemas onde se busca um controle eficiente a um custo acessível. Os microCLP, normalmente contém uma estrutura básica que engloba o processador, memórias e alguns pontos I/O, mas também possui módulos que permitem que outros recursos sejam adicionados. ^(ref.?)

A prioridade no caso de microCLP é garantir que a estrutura básica seja suficiente para o sistema de controle, ou seja, que não seja necessário adicionar outros dispositivos para realizar a programação ou a interface com os componentes do sistema. ^(ref.?)

Quanto ao hardware, em geral, sua estrutura é baseada em processadores operando a até 40Mhz e são muitas vezes utilizados microcontroladores (Swider, 2003). Os recursos de memória são limitados e portanto o tamanho dos programas e números de funções que podem ser incorporados também. Mesmo assim o mercado dos microCLP é o maior por conta de sua facilidade de aplicação com relação aos de maior porte. ^(ref.?)

Devido a estas características é possível, em princípio, desenvolver um microCLP com hardware equivalente ao encontrado no mercado. Um microcontrolador pode ser incorporado como a unidade principal de processamento, facilitando muito o processo de desenvolvimento por incorporar em um mesmo circuito integrado o processador, memórias e interfaces de comunicação.

1.2. Objetivo

construível por empresas como:

O objetivo foi o de projetar e construir um microCLP que atendesse as normas da IEC61131-3 e ~~que tivesse~~ *do trabalho envolve o* um desempenho equivalente aos similares no mercado (Atos, Altus, Siemens, etc). O hardware deveria apresentar características elétricas compatíveis com componentes para automação industrial e o software de programação deveria ser amigável e trabalhar com linguagens conhecidas (Diagrama de Relés e Instruction List).

1.3. Organização do texto

único

O texto é dividido em quatro capítulos, além da introdução. O segundo trata dos fundamentos dos microCLP, ou seja, detalhe-se a estrutura interna, os componentes que fazem parte do dispositivo e a metodologia de processamento convencional. As normas referentes aos CLP também são apresentadas neste capítulo.

capítulo

O terceiro trata da especificação técnica do CLP elaborado, ou seja, uma descrição dos recursos implementados internamente e também a característica da programação utilizada com o microCLP.

O quarto capítulo trata dos detalhes do projeto, ou seja, possibilidades de hardware e como se configura os recursos do microCLP

No quinto capítulo temos a implementação do microCLP, uma aplicação real dos recursos apresentados nos itens anteriores.

Por fim temos as conclusões e as referências do documento.

o capítulo com o

do trabalho

2. FUNDAMENTOS

2.1 Visão geral do MicroCLP

O tipo de CLP utilizado em ambientes industriais ou de qualquer outra aplicação dependem fortemente das características do sistema. A demanda de recursos define o porte e as características de operação do CLP que é aplicado no sistema de automação. (ref.?)

Apesar das diferenças encontradas em tipos específicos de CLP para cada aplicação, podemos apontar alguns componentes que necessariamente fazem parte de qualquer CLP. São eles o módulo de hardware com a unidade de processamento, que contém o processador e memórias e é responsável pela execução das tarefas de controle, a interface de I/O, que realiza a conexão com outros componentes do sistema de controle e a interface de comunicação, que recebe a programação e envia dados e estatísticas do controlador. (ref.?)

Os CLP podem se diferenciar pelas características das interfaces de I/O, tipo de comunicação, funções de controle, etc, mas convencionou-se classificar os CLP pela quantidade de pontos de I/O (Marchesan, 2004). Podemos verificar essa classificação na Tabela 2.1.

Tabela 2.1. Categorias de CLP (ref.?)

Categoria	Número máximo de pontos de I/O
CLP	Acima de 2500 pontos
MiniCLP	Entre 250 e 2500 pontos
MicroCLP	Até 250 pontos

A Tabela 2.1 indica que o microCLP pode conter até 250 pontos de I/O, mas na prática não se encontra controladores com essa quantidade de pontos de I/O. O que se encontra são microCLP com um número menor de pontos de I/O e módulos de hardware que podem ser conectados para aumentar a quantidade de recursos (ref.?)

Como os CLP de categorias superiores, os microCLP costumam ser concebidos para permitir que diferentes tipos de módulos de hardware possam ser agregados ao módulo da unidade de processamento de acordo com a aplicação. (ref.?)

Como os CLP são, em geral, compostos por módulos de hardware, isto é, módulo da unidade de processamento mais os módulos adicionados conforme a necessidade, a diferença da categoria do microCLP para com as outras está na capacidade e recursos disponíveis no módulo da unidade de processamento. Normalmente o módulo dispõe de 8 à 20 ponto de I/O, um display e alguns botões que permitem a programação e configuração diretamente no microCLP e também uma interface de comunicação. (cf.??)

Essa característica permite que ele possa ser aplicado no sistema de controle sem a necessidade de módulos extras (Marchesan, 2004).

2.2 Aplicações do MicroCLP

A seguir são listadas algumas aplicações citadas por fabricantes de microCLP (Carvalho, 2002) (Marchesan, 2004). Pode-se notar que os sistemas citados não constituem linhas completas de produção, são na maioria das vezes controles de máquinas individuais ou sistemas independentes ou pouco integrados a sistemas de supervisão.

- Equipamentos de transporte
 - Controle de esteiras transportadoras
 - Controle de plataformas de elevação
 - Controle de elevadores
 - Controle de silos
 - Controle de alimentadores automáticos

- Instalação elétrica e automação de edifícios
 - Controle de iluminação (exterior e interior)
 - Controle de portas e portões
 - Controle de persianas e toldos
 - Controle de sistemas de irrigação

- Aquecimento/ ventilação/ condicionamento
 - Controle de parâmetros de consumo energético
 - Controle de aquecedores
 - Controle de instalações frigoríficas
 - Controle de sistemas de ventilação
 - Sistemas de ar condicionado

- Máquinas
 - Controle de motores, bombas e válvulas
 - Controle de compressores de ar
 - Controle de sistemas de exaustão e filtragem
 - Controle de estações de tratamento de água
 - Controle de serras e plainas
 - Controle de equipamentos químicos e purificação

- Sistemas de monitoramento operacional
 - Controle de acesso
 - Supervisão de controles de veículos
 - Controle de sistemas de alarme
 - Monitoração de limites de variáveis
 - Controle de sistemas de semáforos
 - Controle de transporte de bagagem

- Soluções especiais
 - Controle de sistemas de energia solar
 - Controle aplicado em embarcações
 - Controle de painéis indicadores e sinalização de tráfego

Como exemplo de microCLP no mercado nacional, ^{tem-se} a linha LOGO! da empresa Siemens é citado como o primeiro da categoria disponibilizado no mercado, ^{além dos} Smart da Metaltex, Zélio da Schneider, Clic da WEG, TICO da Atos e também modelos da Fertron e Altus, ^{etc.}

Muitas vezes os fabricantes costumam nomear o microCLP de relé programável ou relé inteligente ou até microcontrolador programável. ^(ref.?)

CLP DE BAIXO CUSTO PARA SISTEMAS DE AUTOMAÇÃO DE PEQUENO PORTE

2.3 Normas técnicas existentes

Assim como em outros segmentos de produtos eletrônicos, as empresas que fabricam PLCs devem obedecer a normas para que seja possível a compatibilidade dos produtos de modo que favoreça o usuário final que não precisa focar em uma estrutura proprietária de algum fabricante, Viabiliza-se a conquista de novos mercados pelos fabricantes e também facilita-se o desenvolvimento de novas de tecnologia já que os fabricantes devem seguir os parâmetros definidos pela norma. (ref.?)

A principal vantagem do estabelecimento das normas é a redução de custos, principalmente para o usuário final. Com as normas, o desperdício de recursos é reduzido, tanto humano com treinamento, quanto de equipamento com sua manutenção e sua instalação. (ref.?)

As normas trazem também redução da probabilidade de erros, permite a reutilização de software e hardware, facilita a implementação do sistema de controle e permite a intercambialidade entre componentes de controle de diversas empresas. (ref.?)

O conjunto de normas em questão é a IEC61131, que padroniza o funcionamento dos CLP. Ela é dividida em cinco partes:

- **IEC61131-1 –GENERAL INFORMATION**

Estabelece definições e identifica as características relevantes na seleção e aplicação de CLP. Identifica as principais funcionalidades dos CLP.

- **IEC61131-2 – EQUIPMENT REQUIREMENTS AND TESTS**

Especifica os requerimentos dos CLP e relaciona testes para os CLP e seus periféricos.

- **IEC61131-3 – PROGRAMING LANGUAGES**

Define os elementos básicos de programação, a sintática e a semântica para as linguagens de programação mais frequentemente utilizadas. Define os campos de aplicação, os testes e como os fabricantes podem agregar novos componentes. As linguagens definidas são três gráficas: Diagrama de Relés e Diagrama de blocos e SFC; e duas textuais: Lista de Instruções(IL) e Texto Estruturado.

- **IEC61131-4 – USER GUIDELINES**

Documento técnico com informações e parâmetros para a aplicação desta padrões da norma pelo usuário final.

- **IEC61131-5 – MESSAGE SERVICE SPECIFICATION**

Define a comunicação de dados entre o CLP e outros equipamentos usando a Manufacturing Message Specification (MMS) definido no IEC9506.

- **IEC61131-7 – FUZZY CONTROL PROGRAMMING**

Define os elementos de programação de CLP baseados na lógica Fuzzy de controle.

- **IEC61131-8 – GUIDELINE FOR APPLICATION AND IMPLEMENTATION OF PROGRAMMING LANGUAGES**

Guia para desenvolvedores de software para as linguagens de programação de CLP definidas por esta norma.

Em função da natureza do presente trabalho, especial atenção é dada para o item IEC61131-3, que define as técnicas de processamento de controle e também as linguagens de programação que podem ser utilizadas nos CLP.

Esse item trata especificamente das características que permitem a compatibilidade entre dispositivos de diferentes fabricantes, mais especificamente para a programação utilizada. Características elétricas e recursos variam muito de aplicação para aplicação, mas a lógica de controle que deve ser aplicada em geral deve ser a mesma.

2.4 Linguagens de programação da IEC61131-3

Como característica principal, a IEC61131-3 define uma forma de estruturar o programa de controle e também um aspecto muito importante na programação de sistemas complexos, a componentização, ou seja, a capacidade de decompor o programa de controle em partes menores, capazes de serem gerenciadas de forma independente.

Uma consequência da implementação desta estrutura de programação é a exigência de uma interface de programação que permita trabalhar com técnicas de alto nível e que gerencie os componentes que são criados pelo usuário para futura reutilização, e também a exigência de um hardware com recursos de processamento e quantidade de memória adequada, tanto para o programa de controle quanto para as variáveis.

Para implementar o aspecto da componentização no CLP a IEC61131-3 estabelece um método de processamento multitarefa para o processamento do programa de controle.

Na multitarefa, as ações ocorrem disparadas por eventos gerados por mudanças de estado ou por tempo (método preemptivos e método cooperativo) (Caropreso, 2044).

Quanto às linguagens a IEC61131-1 estabelece os seguintes tipos:

Lista de Instruções (IL): linguagem textual de nível próximo a linguagem de máquina, similar ao assembler e baseada no conceito de acumulador simples. Normalmente utilizada em problemas menos complexos ou também como resultado da compilação de uma linguagem mais complexa.

Texto Estruturado (ST): linguagem textual de nível similar ao Pascal, é utilizada para cálculos aritméticos mais complexos pelo maior número de instruções que a Lista de Instruções.

Diagrama de Relés (LD): linguagem gráfica baseada em diagramas elétricos que representam contatos e bobinas, seguindo o modelo da lógica de relés eletromecânicos. Uma versão híbrida conta com blocos funcionais representando contadores, temporizadores, entre outros.

Diagrama de Blocos Funcionais (FBD): linguagem gráfica baseada em diagramas blocos representando as funções e tarefas. O diagrama modela o sistema em termos do fluxo de sinais entre os blocos.

Sequential Flow Chart (SFC): linguagem gráfica baseada na descrição de operações sequenciais, composta por blocos de ações e transições. Muito similar ao conceito de “máquina de estados” e também derivado do modelamento por Redes de Petri (Caropreso,2004).

2.4.1 Diagrama de Relés (LD)

Dentre as linguagens acima, a LD é a mais freqüentemente encontrada na indústria, devido a sua relação com a lógica de relés eletromecânicos, utilizada há muito tempo nesta área.

O diagrama compõe-se de vários “circuitos elétricos” dispostos horizontalmente, alimentados por duas barras verticais nas extremidades. O “circuito” contém contatos como entradas e bobinas como saídas.

Os “circuitos” nada mais são do que sentenças lógicas, onde as operações realizadas com os contatos definem o sinal que aciona a bobina (Carvalho, 2002).

As operações envolvidas são em sua maioria operações booleanas, como AND, OR, NOT. Pela norma também são definidas as utilizações de blocos funcionais que aumentam as funcionalidades da linguagem

Características dos elementos do diagrama

a) Contatos

Representam as entradas do sistema e podem ser de dois tipos, normalmente aberto e normalmente fechado. Não há limite para o número de contatos utilizados.

b) Bobinas

São atribuídas às saídas do sistema. Muda de estado de acordo com o sinal que é aplicado. Os tipos mais comuns são a bobina comum e a bobina inversa.

c) Blocos funcionais

Os blocos funcionais fazem parte dos recursos definidos para promover a programação por módulos reutilizáveis.

Os blocos são concebidos para se adicionar mais funcionalidades ao LD, permitindo o programa manipular operações mais complexas que um simples ON/OFF.

Estão incluídos blocos de temporização, blocos de contadores, blocos de controle de variáveis analógicas, blocos operações aritméticas, blocos de comunicação, entre outros.

Não existe um número de blocos fixo, o desenvolvedor do programa de controle pode criar novos blocos, para executar a operação desejada e depois disponibilizar o bloco em uma biblioteca que possa ser usada na elaboração de outros programas

Exemplo de LD

Na Figura 2.1 temos uma representação de um diagrama com a indicação de seus principais elementos e na Figura 2.2 temos um exemplo com um bloco funcional.

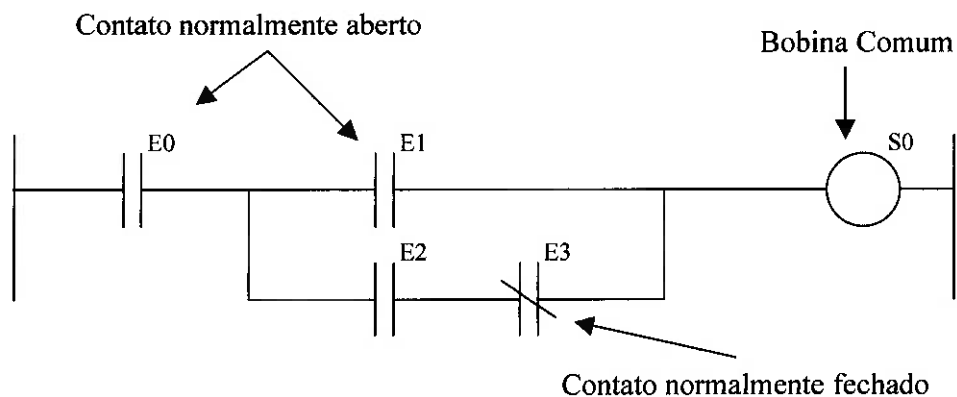


Figura 2.1. LD com a lógica por contatos e bobinas

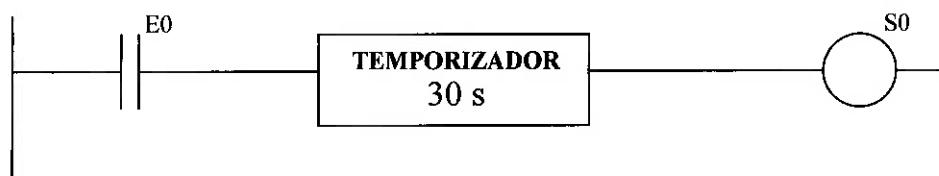


Figura 2.2. LD com bloco de funções

Na Figura 2.1 ~~temos~~ ^{temos} um “circuito” com as seguintes operações: a entrada E3 negada e a entrada E2 fazem uma operação AND, o resultado dessa faz uma operação OR com E1 e por fim uma operação AND com E0. O resultado é colocado na saída S0

Na Figura 2.2 o contato E0 liga a função de temporizador que conta o tempo de 30 segundos antes de acionar a saída S0.

2.4.2. Instruction List (IL)

Essa linguagem é considerada a de mais baixo nível de todas as cinco da IEC61131-3, isso pelo fato de ser uma linguagem textual que contém as instruções que são diretamente executadas pela unidade de processamento do CLP.

As instruções utilizadas para escrever o programa são instruções definidas por uma lista fixa, estabelecida na IEC61131-3. Quanto a estrutura, o texto é escrito com cada instrução em uma linha do programa na ordem em que são tratadas pelo CLP, ou seja, a execução do programa é feita linha por linha no sentido de cima para baixo.

Normalmente a IL é utilizada como resultado da compilação de um programa feito em outra linguagem de alto nível, como as linguagens gráficas. Como as instruções são diretamente entendidas pelo controlador, os elementos das outras linguagens, incluindo até funções, são substituídos por instruções equivalentes durante a compilação. Em alguns CLP, as instruções do processador são as mesmas da IL, o que facilita a execução do programa (Caropreso, 2004).

As instruções básicas do IL incluem operações booleanas como AND, OR, XOR e também outras operações como soma, subtração, comparação de valores. No caso de funções, elas são transcritas na forma de sub-rotinas que são chamadas durante a execução do programa.

Quanto à execução, o processamento do programa trabalha com o conceito de pilha. Assim como na linguagem LD, as linhas de instruções na IL podem compor e combinar diferentes expressões matemáticas. Dessa forma o programa não pode ser executado seguindo somente a sequência de instruções, pois a execução dessas dependem do formato e das operações da expressão.

Para resolver isso a técnica de processamento do programa trabalha com pilhas, onde as instruções e variáveis são ordenadamente armazenadas em pilhas até se encontrar a última operação da expressão e então executa-se as instruções com as variáveis nas pilhas na ordem inversa.

Isso fica mais claro analisando-se o exemplo a seguir. Utilizando a mesma Figura 2.1 e transformando a linha do diagrama para IL, temos:

Tabela 2.2 IL da equivalente ao diagrama da Figura 2.1

	Cód.	Variável	Observações
1	LoaD	E0	Carrega o valor da variável E0 na pilha
2	AND(E1	Operação AND dependente de outras operações
3	OR(E2	Operação OR dependente de outras operações
4	AND	$\bar{E3}$	Operação AND E2 com E3
5)		Executa última operação da pilha
6)		Executa última operação da pilha
7	STore	S0	Atualiza o valor da variável

Na Tabela 2.2 pode-se verificar os códigos que representam as operações da expressão e também pode-se ver a ordem de execução das operações que é definida com o auxílio dos parênteses. Para esta lista de instruções tem-se a seguinte expressão booleana equivalente Eq. (2.1):

$$S0 = (E0 * (E1 + (E2 * \bar{E3}))) \quad \text{Eq. (2.1)}$$

Pela Eq (2.1) pode-se ver as operações que têm preferência durante a execução do código, que é definida pelos parênteses. A operação $E2 \text{ AND } \overline{E3}$ deve ocorrer primeiro, seguida da operação OR com E1 e por fim a operação AND com E0. A execução do IL trabalha da mesma forma.

Como a leitura da tabela de IL segue sempre a ordem de cima para baixo, a pilha define a ordem de execução das operações. No caso do exemplo da Tabela 2.2, o controlador ao ler a linha 1 armazena o valor de E0 na primeira posição da pilha. Ao ler a linha dois, não executa a operação AND por conta dos parênteses, ele guarda o valor de E1 na pilha de variáveis e guarda a operação na pilha de operações.

Na linha três, ocorre a mesma ação, E2 é salvo na pilha das variáveis e a operação OR na pilha das operações. Na linha 4, como a operação AND não tem parênteses ela é executada imediatamente com o valor de $\overline{E3}$ e o valor armazenado na primeira posição da pilha de variáveis, no caso E2.

O valor resultante é então armazenado na primeira posição da pilha de variáveis. Na linha 5, o parênteses faz com que a primeira operação guardada na pilha seja executada entre o primeiro e o segundo valores da pilha de variáveis. O valor é novamente armazenado na pilha de variáveis. O mesmo ocorre para a linha 6.

Por fim, a linha oito faz com que o último valor armazenado na pilha seja enviado para a saída S0 e as pilhas são esvaziadas.

2.5 Arquitetura básica de hardware

Um sistema de automação baseado no uso de CLPs é composto pelo CLP em si, que é o módulo de processamento, com o processador, memórias e algumas interfaces (comunicação, I/O e etc). Ao módulo de processamento estão ligados os módulos de I/O, módulos de comunicação, módulos de interface de usuário, módulos de interface de programação, além dos componentes da máquina em si e do sistema supervisor.

A Figura 2.3 ilustra os componentes que compõem o sistema.

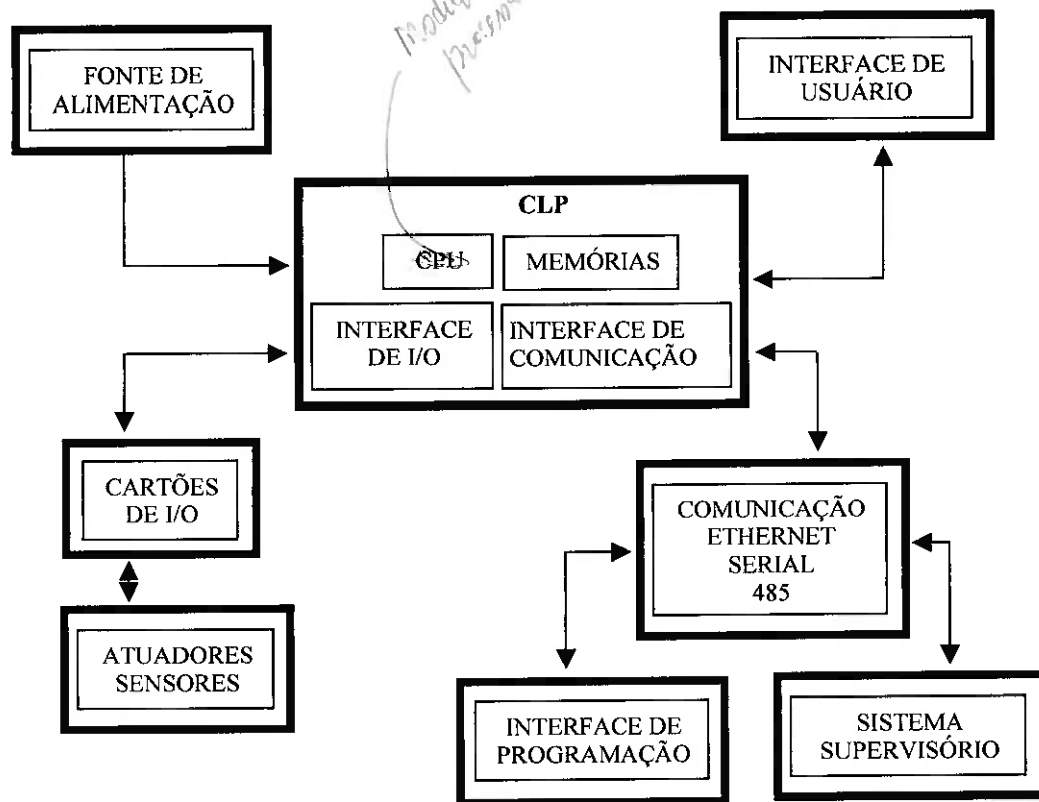


Figura 2.3. Diagrama da arquitetura de hardware envolvendo CLP

Módulo de processamento

gerenciamento Este módulo é a parte principal da arquitetura de hardware, responsável pelo controle de todos os demais componentes e onde são realizadas as operações lógicas do programa de controle. É composta por um processador, memória de programa de controle, memória do firmware e memória de dados do processo.

A memória de programa de controle é a memória manipulada pelo usuário, onde fica armazenado o programa de controle definido pelo usuário e que é executado pelo controlador para executar as tarefas de controle.

A memória do firmware é uma memória que o usuário não tem acesso, ela armazena o programa principal que gerencia o processamento do programa de controle e gerencia as interfaces que fazem parte do sistema.

A memória de dados armazena informações do processo durante a execução do programa de controle. Guarda uma "imagem" do estado das I/O do controlador, variáveis do programa de controle e do próprio firmware.

Interface de I/O

os dispositivos de entrada e saída A interface de I/O é composta pelo circuito responsável por fazer a interface entre o objeto de controle e o controlador. É responsável por passar as informações e executar as ações da maneira correta.

o dispositivo de comando e detecção Nas entradas são conectados sensores, botões, chaves, conversores, etc. Nas saídas são conectados válvulas, atuadores, motores, lâmpadas, etc.

Os sinais elétricos dos componentes externos podem ter diversas características (potência elevada, sinal AC, ruídos, etc). A eletrônica envolvida nos módulos de I/O deve filtrar e transformar o sinal elétrico vindo dos componentes externos para uma configuração que possa ser processada e não danifique o controlador.

Interface de comunicação *do CLP*

Esta interface é utilizada para coletar dados e informações e para programação e alteração dos parâmetros do controlador. Normalmente existem dois tipos de interfaces, uma responsável pela comunicação com outros módulos que fazem parte do sistema e outra voltada para comunicação com o nível de supervisão. A comunicação com outros módulos deve ser feita em tempo real, para se evitar perda de dados entre os dispositivos. São exemplos dessa comunicação os protocolos Fieldbus, Profibus, DeviceNet, etc. Já para a comunicação com níveis superiores de supervisão, tem-se redes de comunicação de aplicação geral e que permitem um maior fluxo de informações, como exemplo as redes Ethernet.

Os recursos de comunicação se tornaram fundamentais para os controladores permitindo não só a programação à distância, mas principalmente o monitoramento *remoto* do processo que está sendo controlado e também a coleta de informações e dados que servem para integração com outros sistemas de controle e também para compartilhar a base de dados com outros níveis do processo produtivo.

Interface de programação

Normalmente essa interface se encontra em um micro-computador, onde o programador pode editar e armazenar os programas de controle desenvolvidos, manipular arquivos e bibliotecas de funções, realizar uma troca do programa que será executado no controlador ou até mesmo monitorar seu funcionamento. Frequentemente utiliza-se das linguagens gráficas para dispor de uma interface amigável e de fácil entendimento para o programador.

Interface de usuário

É uma interface conectada ao módulo de processamento e tem a função de exibir variáveis do processo que está sendo controlado e permite interação direta com o usuário. Basicamente é composta por display, botões e avisos sonoros e luminosos.

Pode ser utilizada como interface de programação do controlador. Neste caso os recursos são limitados e as informações são adaptadas e simplificadas, para serem trabalhadas por meio dos poucos elementos da interface.

Fonte *de energia*

A fonte é responsável por fornecer a energia elétrica para os módulos e interfaces do controlador. Transforma o nível de tensão disponível *na linha de transmissão* *usualmente* em um nível de voltagem e corrente requerido pelo controlador. Num mesmo sistema de controle pode-se ter mais de uma fonte dependendo do porte e da configuração do sistema.

Por exemplo, na indústria o nível de tensão elétrica mais utilizado para sistemas de controle é o 24V DC.

2.6 Funcionamento do CLP

A rotina principal do CLP convencional que executa o programa de controle, nada mais é do que uma varredura seqüencial das instruções desse programa em um "loop" (Carvalho, 2002).

Nesse "loop", primeiramente é feita uma leitura das entradas do CLP, criando uma imagem das entradas através da interface de *entradas* *criando uma imagem* I/O que servirá de informação para a execução das instruções. Na seqüência, o módulo de processamento utiliza a imagem das entradas para executar as instruções do programa de controle. Ao final, como resultado, temos *processadas as* uma *nova* imagem das saídas do sistema. Por fim, a nova imagem das saídas é atualizada na interface de *saídas* *(0)* I/O.

O tempo total gasto em cada ciclo deste loop é chamado de tempo de varredura, depende de alguns fatores como a velocidade de processamento, o tamanho do programa de aplicação e da quantidade de pontos de entrada e saída.

onde está a explicação do firmware?

3. ESPECIFICAÇÃO TÉCNICA DO MICROCLP

3.1. Hardware

O processador escolhido para o módulo de processamento do microCLP é um microcontrolador que contém internamente recursos de comunicação e memória, o que facilita o projeto do circuito eletrônico.

A escolha pela utilização do microcontrolador limita a quantidade disponível de memória para o programa de controle e para os dados, mas tem-se um componente que combina três recursos fundamentais, economizando espaço no módulo de processamento, contribuindo para reduzir as dimensões do microCLP. O que é uma característica reconhecida como um aspecto positivo para esse tipo de controlador.

Outra vantagem do microcontrolador é a disponibilidade de ferramentas de desenvolvimento e documentação.

O microcontrolador escolhido para este trabalho foi um microcontrolador da empresa Microchip, que tem disponível uma vasta documentação e exemplos de aplicação. A programação deste microcontrolador pode ser feita com um programador “caseiro”, ou seja, não necessita de um dispositivo de programação da própria empresa Microchip. O software de programação do firmware em linguagem assembler também é distribuído gratuitamente. Atualmente, a disponibilidade desse componente no mercado é relativamente grande se comparado aos de outras empresas e o preço é mais um atrativo frente a microcontroladores de outras empresas como Motorola, Atmel, Rabbit.

O microcontrolador escolhido foi o PIC18F452, principalmente pelo número de I/O e pela velocidade de processamento.

Abaixo temos um resumo das características deste microcontrolador:

Tabela 3.1. Recursos do PIC18F452 (Microchip, 2001)

Features	PIC18F452
Operating Frequency	DC - 40 MHz
Program Memory (Bytes)	32K
Program Memory (Instructions)	16384
Data Memory (Bytes)	1536
Data EEPROM Memory (Bytes)	256
Interrupt Sources	18
I/O Ports	Ports A, B, C, D, E
Timers	4
Capture/Compare/PWM Modules	2
Serial Communications	MSSP, Addressable USART
Parallel Communications	PSP
10-bit Analog-to-Digital Module	8 input channels
RESETS (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect	Yes
Programmable Brown-out Reset	Yes
Instruction Set	75 Instructions
Packages	40-pin DIP 44-pin PLCC 44-pin TQFP

Com esse microcontrolador tem-se um “clock” que pode chegar a 40MHz e também uma memória “flash” de 32Kbytes como principais características. A memória ~~“eeprom”~~ ^{“EEPROM”} tem apenas 256 bytes, assim o programa de controle carregado no microCLP ocupa também parte da memória “flash” que também guarda o “firmware” do microcontrolador.

O uso da memória “flash” para o programa do usuário é possível pois este microcontrolador oferece um acesso facilitado a esta memória por meio de instruções específicas do processador, permitindo que sejam mantidas tabelas de dados em qualquer posição da memória.

Uma vantagem que a memória “flash” tem é que sua leitura é mais rápida que a ~~“eeprom”~~ e mesmo sendo a mesma memória do “firmware” não é necessário nenhum recurso especial, como um programador, para alterá-la. O “firmware” continua inalterado e apenas a tabela com o programa de controle é alterada.

3.1.1 Componentes do MICROCLP

Como podemos ver na Tabela 3.1, ~~temos~~ ^{tem-se disponível} um número de 36 pinos de I/O, Tem-se disponível também entrada analógica e interface serial que também são utilizados pelo microCLP.

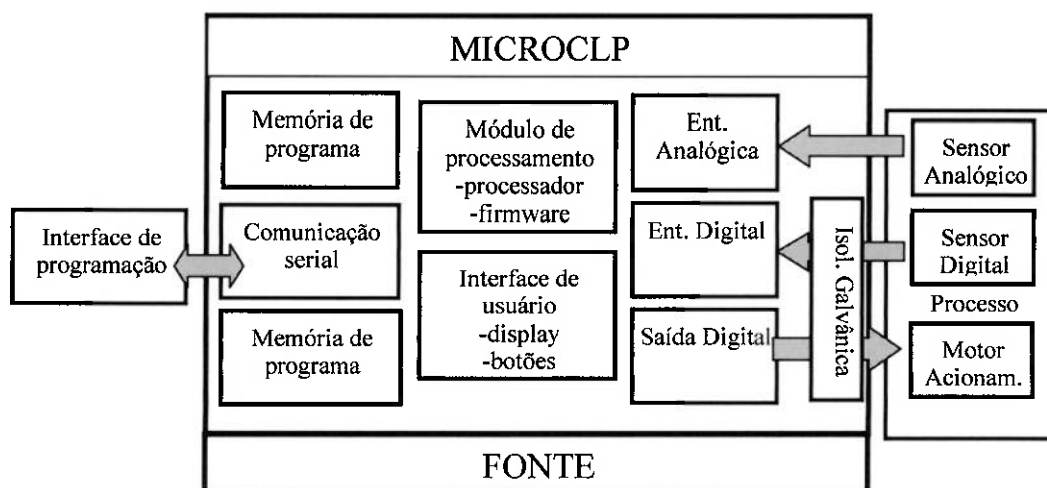


Figura 3.1. Diagrama com os componentes do microCLP

Na Figura 3.1 tem-se um diagrama com os componentes implementados e como é a arquitetura do microCLP. ~~Temos~~ ^{tem-se} uma estrutura, englobando o módulo de processamento, com as memórias e os circuitos de entrada e saída. Junto com ele, temos os componentes que foram agregados a mesma estrutura, no caso os circuitos de potência para a interface de I/O com a devida isolação galvânica com relação aos componentes externos. Temos também o driver da comunicação serial e os circuitos para a interface de usuário com os botões, display, buzina e LEDs.

Estrutura física

As placas de circuito impresso dos módulos do microCLP, foram feitas de acordo com um formato que permitisse que elas pudessem ser encaixadas em uma “caixa” convenientemente selecionada.

A “caixa” selecionada tem um tamanho padrão, própria para ser encaixada em painéis elétricos por meio de trilhos DIN. A “caixa” é a MODULBOX H53 da empresa ITALTRONIC, com o comprimento de 105 milímetros, suficientes para comportar a interface de usuário, com display e botões, na parte superior.

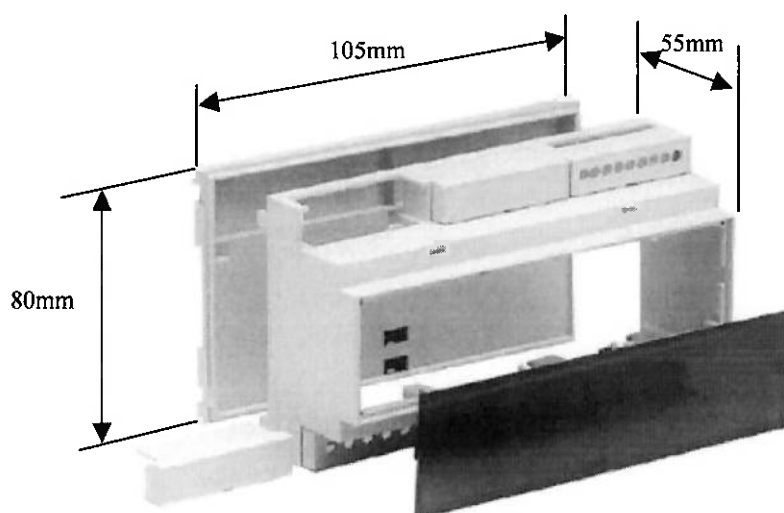


Figura 3.2. Caixa do modelo H53 da ITALTRONIC em seu formato original

A caixa da Figura 3.2, possui espaços internos onde podem ser encaixadas as placas de circuito impresso.

No presente trabalho, o circuito eletrônico completo foi dividido em duas placas, uma placa é chamada de “driver” de potência, por conter o regulador de voltagem para o nível de 5 volts exigido pelo microcontrolador e por conter os circuitos que convertem os níveis de energia dos sinais do módulo de processamento para a interface de I/O e da entrada analógica.

A outra placa é chamada de placa de controle, pois contém o módulo de processamento e todos os componentes da interface de usuário, como os LEDs de status das I/O, conexão para o display de caracteres, a buzina e os botões de configuração e de reset.

Características elétricas

O microCLP foi projetado considerando-se as aplicações em automação eletro-pneumática, onde opera-se os dispositivos com uma voltagem de 24V DC. O microCLP desenvolvido pode operar com uma fonte com um nível mínimo de 20V DC e um nível máximo de 30V DC. A figura 3.2 mostra um exemplo simples de uma fonte de 24V DC.

A voltagem da fonte é convertida para 5V DC que é a voltagem utilizada pela placa de controle, isso é feito por um regulador linear de voltagem.

Saídas digitais

As saídas digitais são compostas por 8 micro-relés acionados com 24V DC de entrada, e podem suportar na saída até 500mA com 128V AC ou 1A com 30V DC.

As saídas do microCLP fornecem uma voltagem igual a voltagem fornecida pela fonte quando estão no nível lógico 1 e operam em aberto quando estão no nível lógico 0. A figura 3.3 ilustra essa característica.

A frequência de operação máxima recomendada pelo manual do modelo de micro-relé utilizado é de 10Hz.

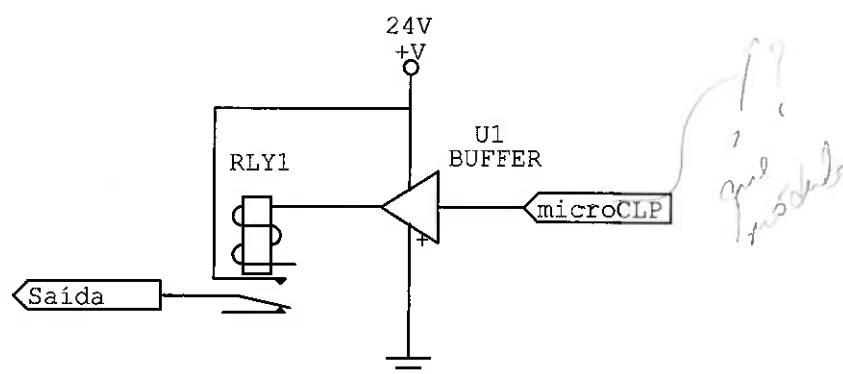


Figura 3.3. Circuito do “driver” de potência para a saída digital

Entradas digitais

A entrada digital é isolada por meio de um opto-acoplador que separa galvânicamente a placa de controle do nível de tensão dos componentes externos. Externamente existe uma conexão para cada entrada, e da entrada para o opto-acoplador tem-se um resistor para limitar a corrente. A entrada pode ser acionada com o nível lógico 1, com a voltagem variando de 18 até 30V DC. A Figura 3.4 ilustra o circuito.

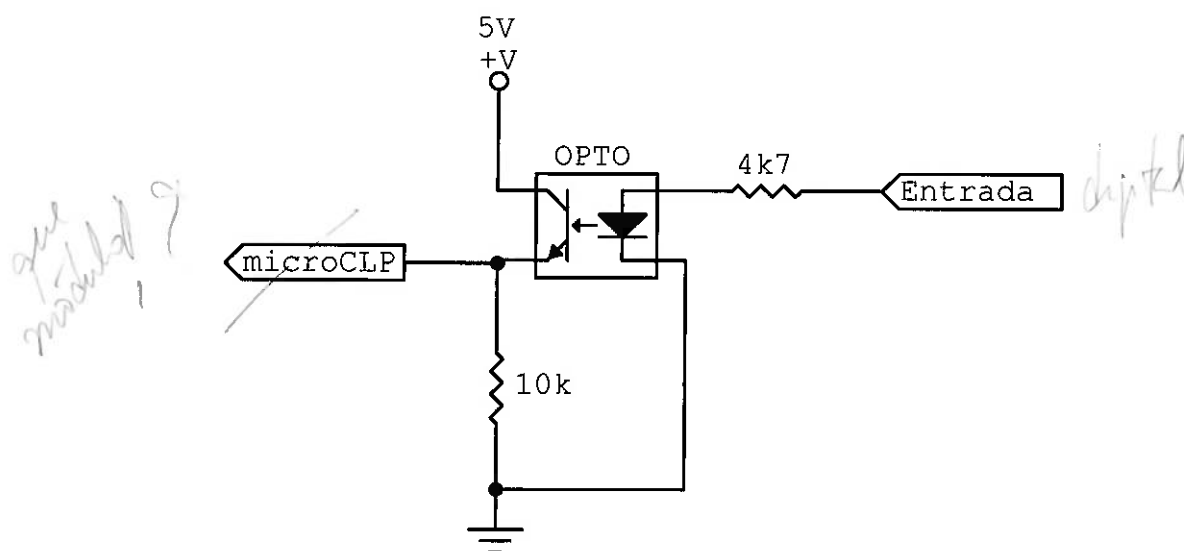


Figura 3.4. Circuito da entrada digital do microCLP

Entrada analógica

Sensores analógicos para aplicações industriais normalmente estão disponíveis em duas configurações principais normalizadas para este tipo de aplicação. Um tipo gera na saída um sinal de tensão de 0V à 10V DC e outro tem como saída um sinal de corrente de 0,4mA à 0,20mA.

O microcontrolador tem um recurso de entrada analógica que pode variar de 0V à 5V DC nos seus pinos. Isso inviabiliza uma conexão direta da saída do sensor com o microcontrolador. Assim, a interface de I/O dispõe de dois circuitos disponíveis na placa de “driver” de potência onde o usuário pode escolher o circuito de acordo com a saída do sensor.

Para o sinal em tensão tem-se um divisor de tensão com dois resistores de 10k ohms e para o sinal em corrente temos um resistor de 250 ohms para a conversão do sinal para 5V DC no máximo.

Na placa de “driver” de potência, as duas opções são selecionadas por meio de “jumper”, mas ainda é possível inserir circuitos mais complexos que os dois tipos apresentados acima. Para isso a placa contém um conector tipo barra de pinos para se adicionar um circuito extra.

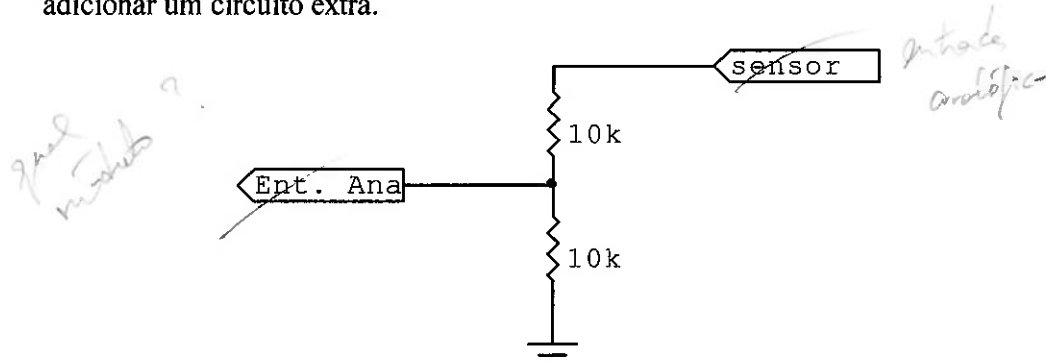


Figura 3.5. Circuito para sensores com sinal em tensão de 0 à 10 V DC

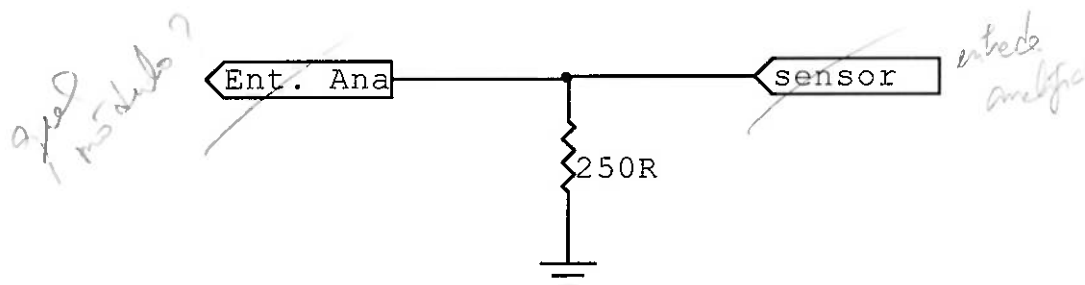


Figura 3.6. Circuito para sensores com sinal em corrente de 4 à 20 mA

Ambos circuitos das Figuras 3.5 e 3.6, são relativamente simples e não exploram a precisão que o microcontrolador pode operar. Entretanto, se necessário, pode-se efetuar uma correção do sinal por software e uma calibração prévia do sinal do sensor. O circuito da Figura 6 é um divisor de tensão que reduz a variação de 0 à 10V para 0 à 5V. O circuito da Figura 7, o sinal que vai para o microcontrolador é a tensão sobre o resistor de 250 Ω . Esse circuito é o que apresenta uma maior perda na precisão, já que a variação ideal seria de 0 à 5V, mas com esse circuito consegue-se apenas uma variação de 1 à 5V.

Interface de usuário

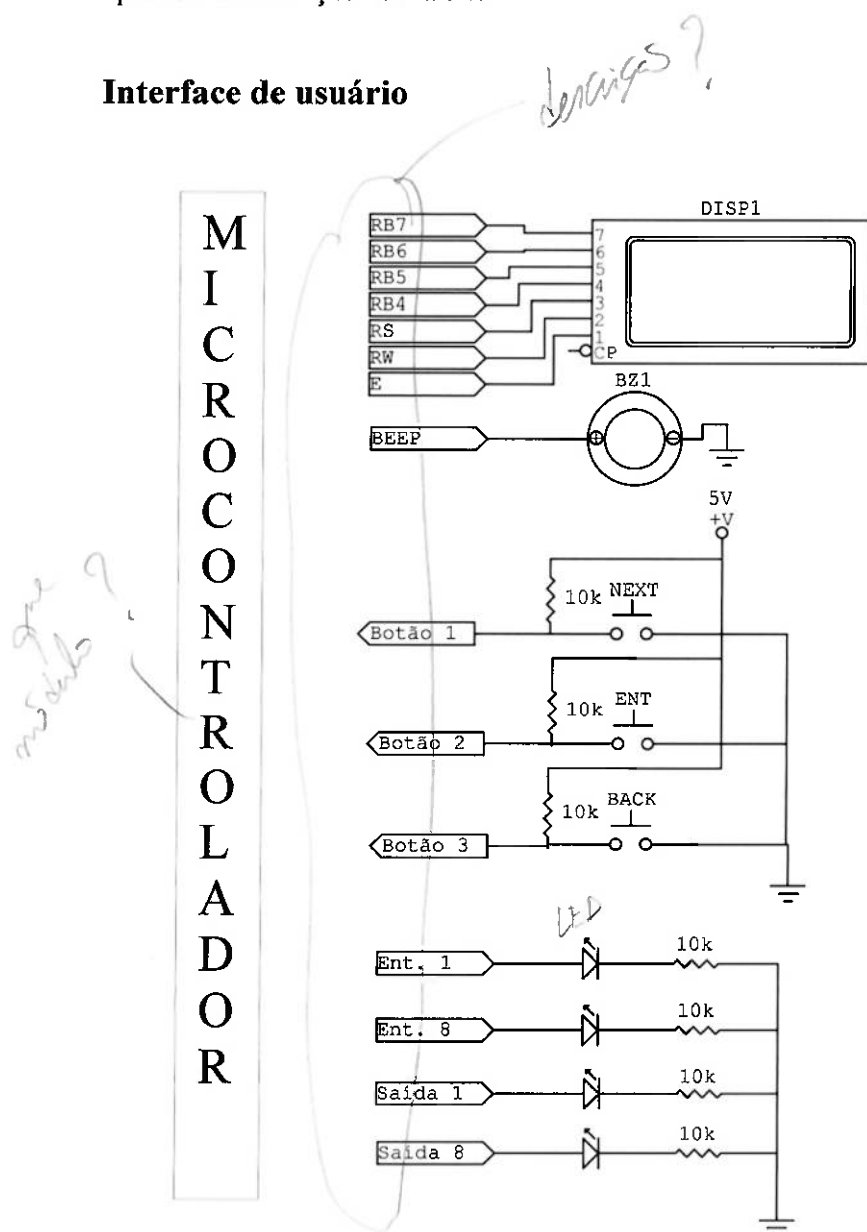


Figura 3.7. Circuito equivalente a interface de usuário

A interface é composta por dispositivos de monitoração como o display e a buzina e dispositivos de comando como os três botões para a interação com o menu de opções de operação do microCLP que é mostrado no display

Comunicação serial

Como o microcontrolador já tem intrnamente recursos para uma comunicação serial que opera em níveis TTL, a interface de comunicação serial contém apenas um componente que converte o sinal do nível TTL para os níveis do padrão RS232, o que permite que o microCLP se conecte a interface serial de computadores ou outros dispositivos. O circuito da interface está ilustrado na Figura 3.8.

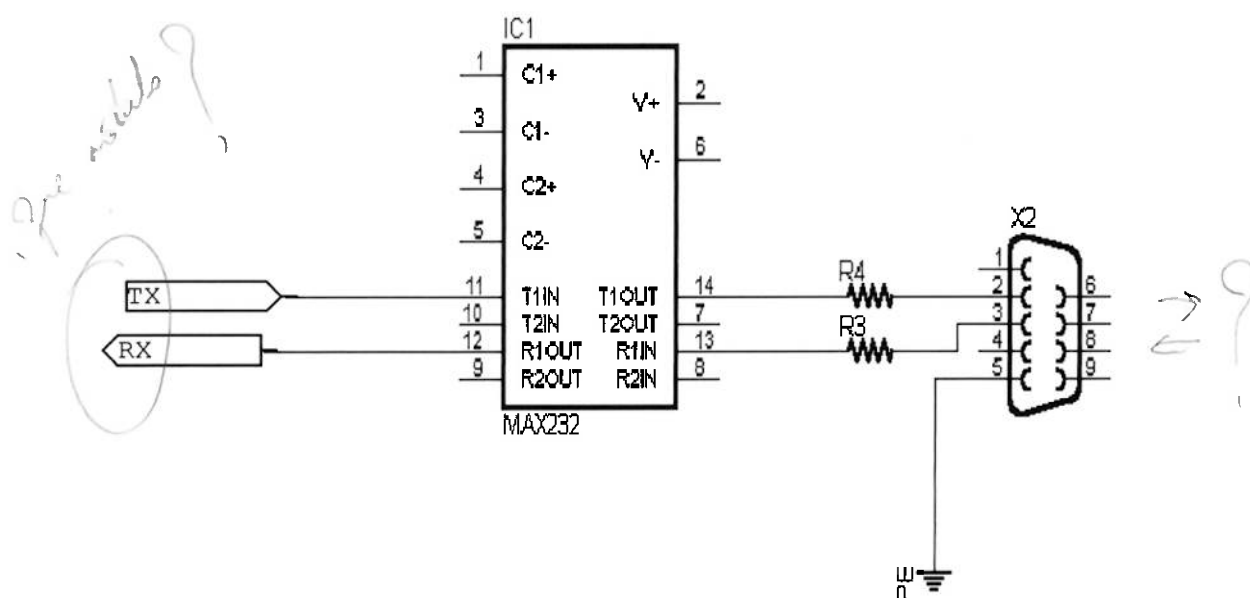


Figura 3.8. Circuito equivalente ao “drive” da comunicação serial

3.2 Linguagens de programação

As linguagens de programação que foram implementadas para o microCLP são a linguagem gráfica LD e a linguagem textual IL, apresentadas nas seções 2.4.1 e 2.4.2 respectivamente.

A proposta é que o programa de controle será desenvolvido pelo usuário em LD numa interface de programação. Após sua edição, esta interface compila o programa de controle para IL e para os códigos que o microCLP entende, que serão depois carregados no microCLP.

3.2.1 LD para o microCLP

Os componentes desta linguagem adaptados para o microCLP são compostos pelos contatos normalmente aberto, o contato normalmente fechado, a bobina e os blocos funcionais.

A estrutura do diagrama é a mesma apresentada na seção 2.4.1, onde temos os contatos arranjados de forma a representar uma expressão matemática, que acionam as bobinas com o resultado da expressão.

Os contatos podem ser representações dos sinais, mas também podem representar o estado de uma bobina (sinal de saída do controlador) ou então representar bobinas virtuais, ou seja, apenas representam variáveis internas.

Para o microCLP tem-se as seguintes variáveis, apresentadas na tabela 3.2, que podem ser usadas no diagrama de relés:

Tabela 3.2. Variáveis que podem ser utilizados na programação

Entradas	E0 – E7
Saídas	S0 – S7
Saídas virtuais	V0 – V7

Uma atenção especial deve ser dada aos blocos funcionais, que para o microCLP são previamente definidos apenas três. São eles o temporizador, o contador e a entrada analógica.

Os blocos têm como saída, sinais equivalente a contatos, mas cada bloco têm parâmetros e sinais que os habilitam ou os reinicializam. Esses sinais são representados por bobinas que também são inseridas no diagrama.

Bloco temporizador

O bloco temporizador manipula a ação de um contato, por exemplo, fechando-o por um certo tempo pré-definido ou de outras formas. Por conta disso, define-se quatro maneiras que o temporizador atua num contato:

1. Fecha contato por um certo tempo e depois abre o contato
2. Abre o contato por um certo tempo e depois fecha o contato
3. Espera um certo tempo e depois abre o contato
4. Espera um certo tempo e depois fecha o contato

Além destas configurações a contagem pode ser em décimos de segundo, segundos ou minutos.

Como entrada tem-se um sinal de bobina que habilita o bloco e o sinal de bobina que limpa a contagem caso o bloco esteja habilitado.

Na Figura 3.9, tem-se um esquema do bloco temporizador e na Figura 3.10 um circuito de exemplo de aplicação do bloco temporizador.

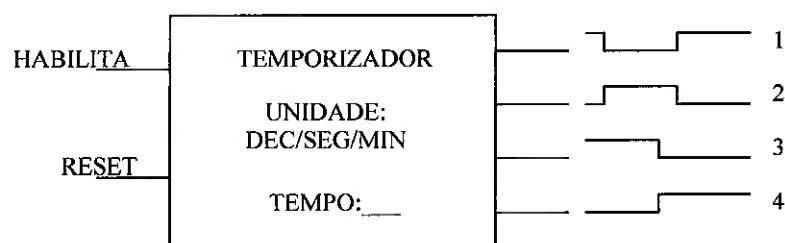


Figura 3.9. Esquema com as propriedades do bloco temporizador.

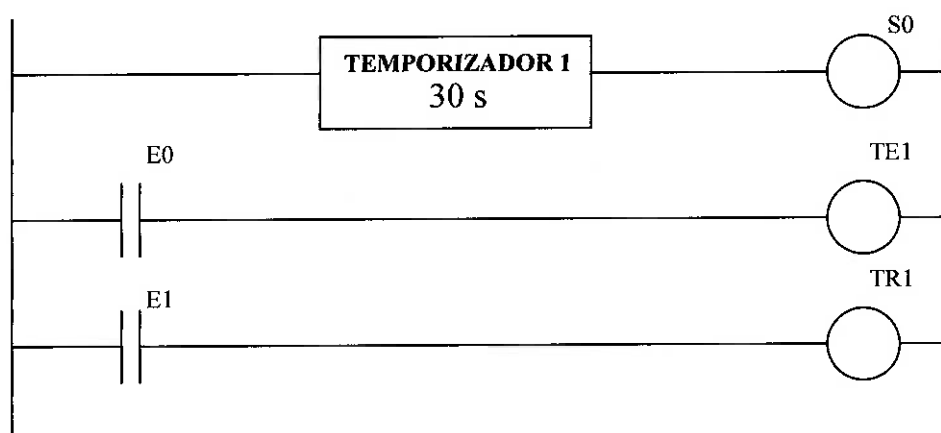


Figura 3.10. Aplicação do bloco temporizador.

Na Figura 3.10, tem-se a bobina TE1 representando o sinal que habilita o bloco temporizador e a bobina TR1 o sinal que limpa a contagem do bloco.

A Tabela 3.3 apresenta os parâmetros de programação do bloco temporizador

Tabela 3.3. Resumo dos parâmetros do bloco temporizador

Configuração	Parâmetros	Códigos(decimal)
Tempo	1 byte, valor de 0 à 255	0 - 255
Unidade	0 - segundos	48
	1 - décimos	49
	2 - minutos	50
Tipo de saída	0 – fecha o contato e depois abre	48
	1 – abre o contato e depois fecha	49
	2 – espera o tempo e abre o contato	50
	3 – espera o tempo e fecha o contato	51

Bloco contador

Da mesma forma que o bloco temporizador, o bloco contador manipula a ação de um contato. Para esse tem-se apenas duas possibilidades:

1. Fecha contato após contador atingir a igualdade
2. Abre o contato após contador atingir a igualdade

Quanto às entradas do bloco contador tem-se o sinal que incrementa o contador e o sinal que zera a contagem

A configuração inclui ainda a escolha se a contagem será incrementada pela borda de subida ou na borda de descida do sinal que incrementa o contador.

Na Figura 3.11 tem-se um esquema do bloco contador e na Figura 3.12, um circuito de exemplo de aplicação do bloco.

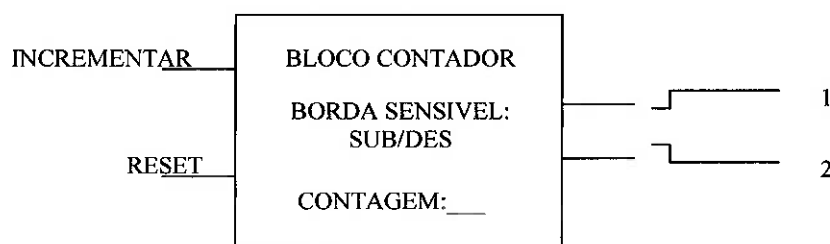


Figura 3.11. Esquema com os parâmetros do bloco contador

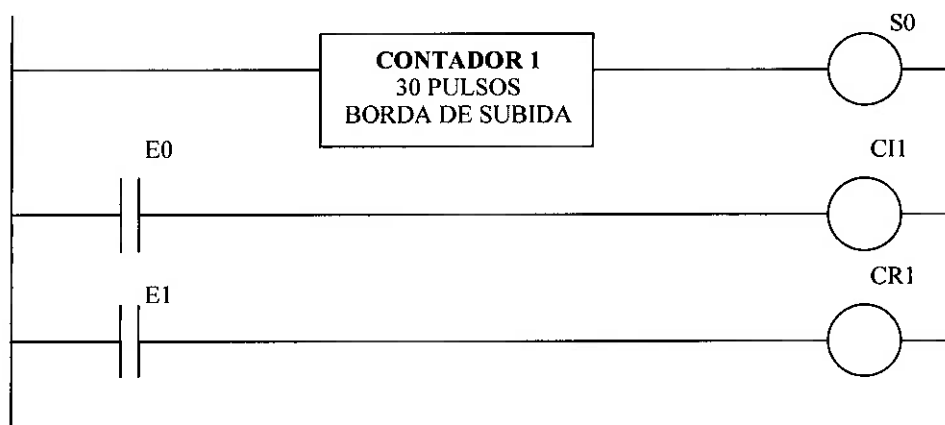


Figura 3.12. Aplicação do bloco contador

A aplicação do bloco contador da Figura 3.12, mostra a bobina C11 que é acionada pelo contato E0 e que incrementa o contador do bloco. O contato E1 aciona a bobina CR1 que representa o sinal que reinicializa o bloco do contador.

A Tabela 3.4 apresenta os parâmetros do bloco contador

Tabela 3.4. Resumo dos parâmetros do bloco

Configuração	Parâmetros	Códigos(decimal)
Contagem	1 byte, valor de 0 à 255	0 - 255
Borda	0 – borda de subida	48
	1 – borda de descida	49
Tipo de saída	0 – fecha o contato	48
	1 – abre o contato	49

Bloco entrada analógica

O bloco de entrada analógica, assim como o bloco contador, tem dois tipos de saída:

- 1.Fecha contato após a entrada analógica atingir a igualdade com valor de referência
- 2.Abre o contato após a entrada analógica atingir a igualdade com valor de referência

Quanto às entradas deste bloco tem-se o sinal que habilita o bloco e a bobina que zera o bloco.

Neste bloco é necessário definir o valor de referência da entrada analógica. O microcontrolador oferece um conversor A/D com precisão de 10bits, mas para este projeto utilizam-se apenas os oito mais significativos, formando um byte. Assim, o papel do conversor é transformar o sinal analógico para um valor variando de 0 a 255 em decimal.

O valor de referência selecionado também deve estar entre o 0 e 255, a escolha depende da análise do sistema e sensor utilizado pelo usuário.

Outro parâmetro a ser selecionado é se a comparação do valor de referência com o valor lido no conversor A/D deve ser realizada quando o último cruza com o valor selecionado indo de um valor superior para inferior, ou o contrário. Isso evita o problema de só comparar o valor exato, já que muitas vezes a mudança do valor na leitura pode ser brusca e em função do “clock” do microcontrolador, o valor lido ultrapassa o valor de referência sem igualá-lo.

Na Figura 3.13 tem-se um esquema do bloco entrada analógica e na Figura 3.14, um circuito de exemplo de aplicação do bloco.

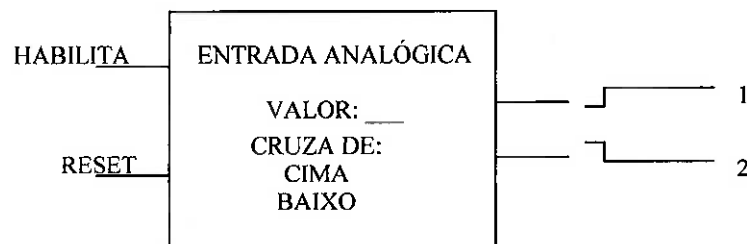


Figura 3.13. Esquema com as propriedades do bloco entrada analógica

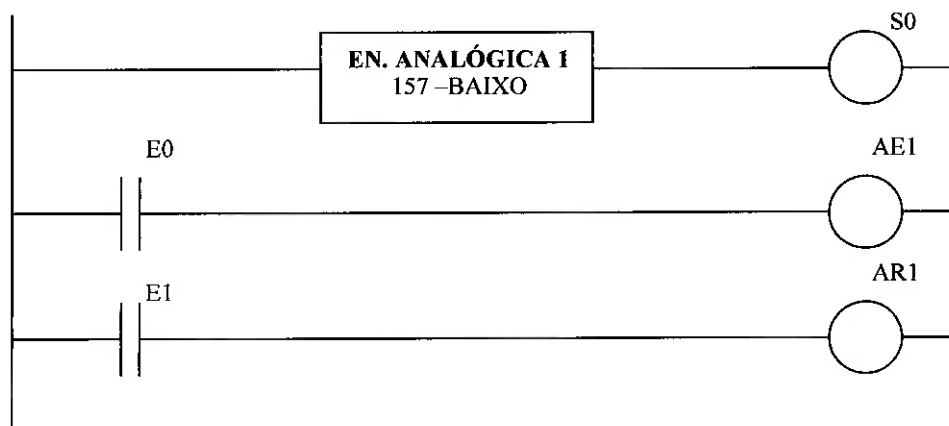


Figura 3.14. Aplicação do bloco entrada analógica

Na Figura 3.14 o bloco entrada analógica é habilitado pelo contato E0 e reinicializado pelo contato E1.

Quanto aos parâmetros, tem-se o valor de referência e o sentido em que o valor lido na entrada analógica deve ser comparado com o valor de referência. Para esse caso, quando o valor lido foi maior ou igual a 157 o bloco aciona a saída.

Tabela 3.5. Parâmetros de programação do bloco entrada analógica

Configuração	Parâmetros de programação	Códigos(decimal)
Valor esperado	1 byte, valor de 0 à 255	0 - 255
Direção	0 – cruza subindo	48
	1 – cruza descendo	49
Tipo de saída	0 – fecha o contato	48
	1 – abre o contato	49

3.2.2 IL para o microCLP

Para o microCLP desenvolvido, a linguagem IL é usada apenas como resultado intermediário da compilação do programa feito em LD.

O programa de controle em IL é importante, pois o microCLP executará o programa de controle como se executa uma IL.

As instruções utilizadas e os códigos equivalentes estão na Tabela 3.6.

Tabela 3.6. Instruções e códigos do IL para o microCLP

Instrução	Modificadores	tipo	Explicação
LoaD	N	booleana	Carrega um valor das variáveis para pilha
STore	N	booleana	Armazena o valor da pilha em uma variável
AND	N, (booleana	Operação booleana AND
OR	N, (booleana	Operação booleana OR
ADD	(inteiro	Operação de Adição
SUB	(Inteiro	Operação de Subtração
GE	(inteiro	Operação de comparação maior ou igual
EQ	(Inteiro	Operação de comparação igual
LE	(inteiro	Operação de comparação menor ou igual
)			Executa última operação carregada na pilha
CALL		nome	Chama uma função
RETURN			Retorna da função

3.3 Execução do Programa de controle

A técnica empregada para execução do programa de controle é a usualmente encontrada em CLPs convencionais, ao se iniciar a execução do programa de controles, o microCLP faz uma imagem das entradas digitais, das saídas digitais, das saídas virtuais e das variáveis de todos os blocos funcionais, guardando os valores em variáveis específicas, então ele executa as instruções armazenadas na memória do programa de controle e depois atualiza a imagem de todas as variáveis, por fim a imagem das saídas é atualizada nas saídas do controlador.

A Figura 3.15 ilustra esse processo.

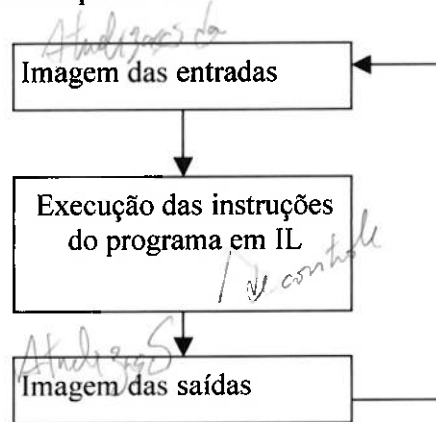


Figura 3.15. Loop de execução do programa de controle

Esse tipo de processamento de controle tem as seguintes vantagens:

- fácil entendimento(intuitivo);
- execução seqüencial das instruções, como deve ser a execução da IL;
- poucas variáveis adicionais, isto é, as variáveis da execução são, na maioria, para manipular as variáveis do programa de controle;

Quanto as desvantagens tem-se:

- atraso na atualização das saídas, como as saídas são atualizadas só no fim de cada ciclo, quanto maior o programa maior o atraso
- por conta do ciclo de execução se a entrada gerar um pulso relativamente rápido o controlador pode não identificar o pulso.
- não permite desvios na execução.

3.4. Firmware do microCLP

Na Figura 3.16 é apresentado um diagrama de como o ciclo de varredura foi implementado no firmware do microCLP. No diagrama é possível identificar três componentes: o programa principal (firmware), a memória do programa de controle e a memória das variáveis de entrada e saída utilizadas pelo programa de controle.

Como descrito anteriormente na descrição da linguagem IL, a execução do programa de controle trabalha com o conceito de pilhas devido as relações de prioridade entre as instruções.

O firmware do microCLP deve, assim, manipular a posição das pilhas através de apontadores que indicam a posição em que a última instrução ou variável foi armazenada. No caso da pilha de variáveis, a primeira posição é sempre utilizada por se tratar do resultado geral das operações que estão sendo executadas. O preenchimento das outras posições da pilha de variáveis e da pilha de instruções ocorrem com o identificação do sinal “(“ junto a instrução, o que denota que uma relação de prioridade é necessária.

Somente a instrução “)” dispara a execução de uma instrução que está na pilha. Após a execução tanto a pilha de instruções quanto a pilha de variáveis são decrementadas.

Quanto ao programa de controle em IL, o firmware deve primeiro ler o cabeçalho do programa de controle, que não faz parte da execução direta do código. Esse cabeçalho contém somente as configurações dos blocos, que são lidos somente uma vez no início da execução. Cada configuração do cabeçalho é direcionada para as respectivas variáveis e definem os parâmetros durante a execução. Após o cabeçalho, o firmware atualiza as variáveis do microCLP e então executa as instruções do programa de controle.

É importante ressaltar que o programa de controle está dividido em blocos, onde cada bloco representa uma equação. Esses blocos são equivalentes a cada linha do programa em LD.

A atualização das variáveis somente é feita após serem executados todos os blocos do programa de controle, ou seja, se um bloco anterior atualizar o valor de uma saída “S0”, o bloco posterior não vai utilizar o novo valor de “S0” e sim o valor presente na imagem deste feita no início do ciclo.

A figura 3.16 ilustra o funcionamento do firmware do microCLP.

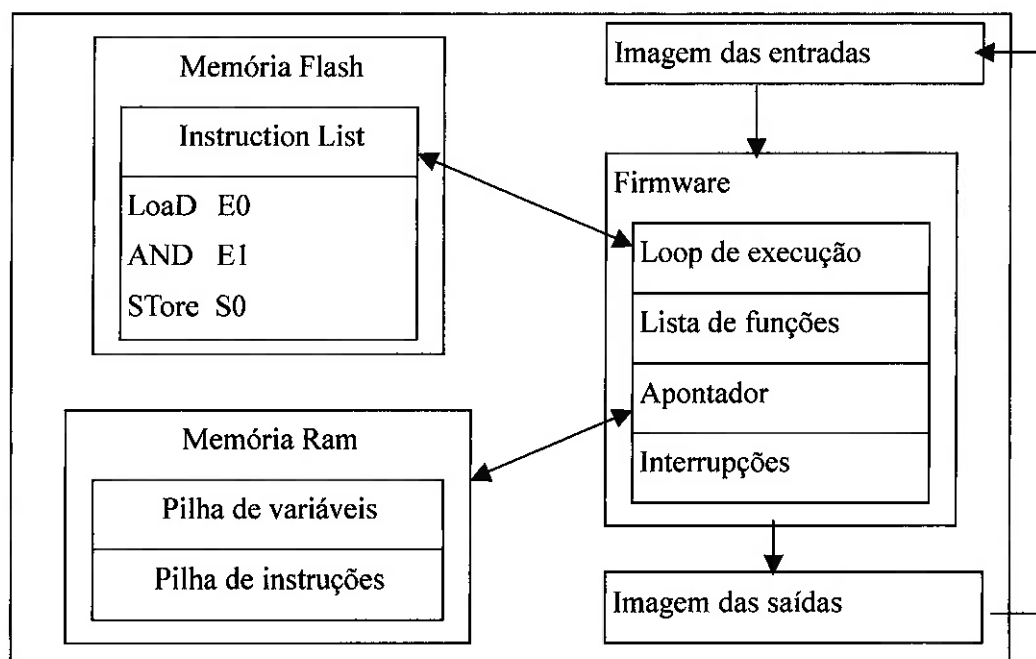


Figura 3.16. Funcionamento do firmware do microCLP

3.5. Interface de programação

A interface de programação, utilizando a linguagem LD para editar o programa de controle, é uma maneira relativamente mais fácil e mais intuitiva para realizar a programação do microCLP.

Em situações específicas pode-se utilizar como interface de programação do microCLP, displays, botões e até painéis IHM. No entanto para casos gerais, utiliza-se uma interface de programação baseada em micro-computadores tipo PC. Isso permite utilizar recursos extras que em geral o controlador sozinho não oferece. Recursos como interface visual, teclado alfanumérico, mais capacidade de armazenamento, manipulação de arquivos em grande quantidade e monitoração.

Para o microCLP desenvolvido, foi assim implementado uma interface de programação compatível com o sistema operacional Windows e com os recursos necessários para realizar a programação do microCLP por meio da linguagem gráfica LD.

O resultado é uma interface para programação possibilita a edição, o carregamento e armazenagem dos programas de controle elaborados pelo usuário em LD.

A Figura 3.17 ilustra a tela de trabalho desta interface de programação.

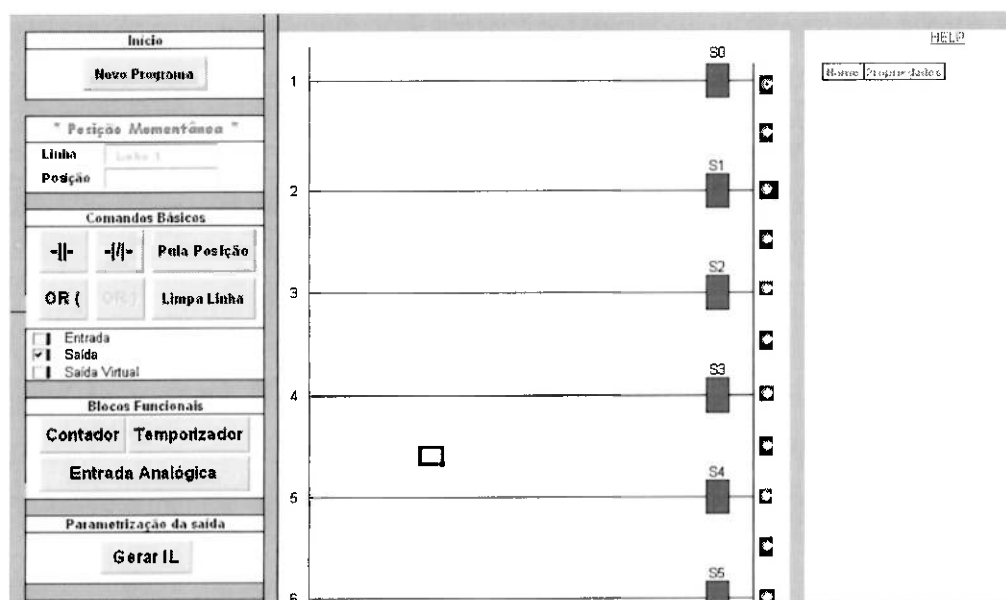


Figura 3.17. Tela de trabalho da interface de programação

A tela de programação foi desenvolvida em linguagem Visual Basic for Applications, dentro do programa Excel, o que permite trabalhar mais facilmente com tabelas de dados. Assim a armazenagem da lista de instruções do programa e da tabela de bytes equivalentes ao programa em IL podem ser salvas e carregados como tabelas com extensão “.xls”.

A figura 3.18 ilustra a tela de trabalho da interface de programação durante a edição de um programa de controle.

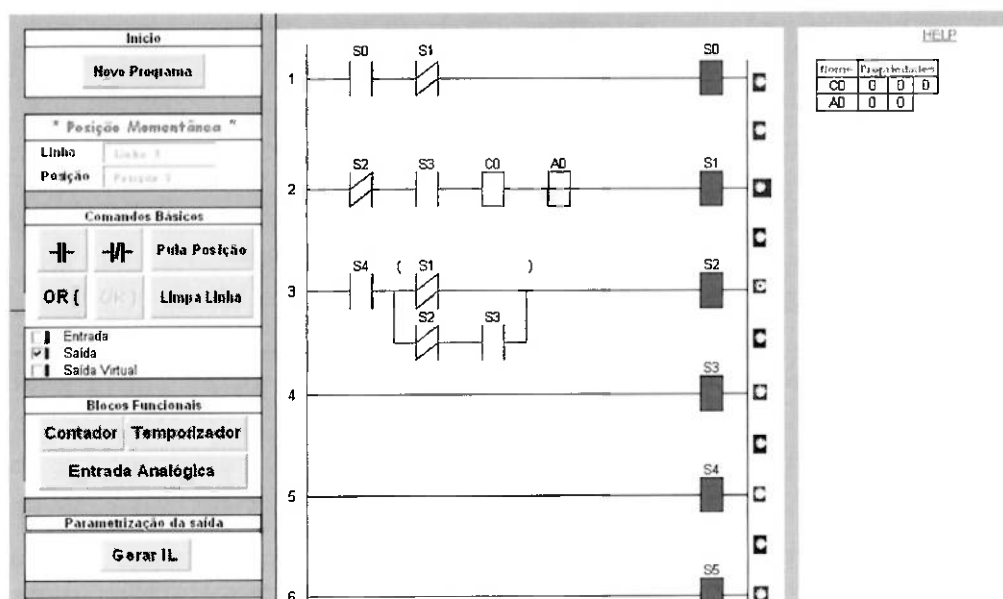


Figura 3.18. Tela de trabalho da interface de programação com um exemplo de programa de controle

4. IMPLEMENTAÇÃO DO MICROCLP

4.1 Estrutura Física

A seguir tem-se algumas fotos, mostrando a visão externa do dispositivo e as placas eletrônica internas com seus respectivos componentes.

As Figuras 4.1 e 4.2 ilustram como a “caixa” original foi trabalhada para abrigar as placas eletrônicas, distribuir espaço para a interface de usuário e permitir o acesso às conexões com os dispositivos externos.



Figura 4.1. Foto da aparência externa do microCLP

Na Figura 4.2, é possível identificar do lado direito da caixa, a o conector DB-9 fêmea que permite a ligação do cabo de comunicação serial do microCLP e as oito saídas digitais.



Figura 4.2. Detalhe das laterais do microCLP

A Figura 4.3 mostra as duas placas de circuito impresso que compõem o hardware do microCLP. Na parte superior está a placa de controle, que contém o microcontrolador, a conexão com o display, a conexão com o programador do microcontrolador, os botões da interface de usuário, uma buzina e os LEDs de monitoramento das I/O. Um cabo flexível faz a conexão entre as duas placas.

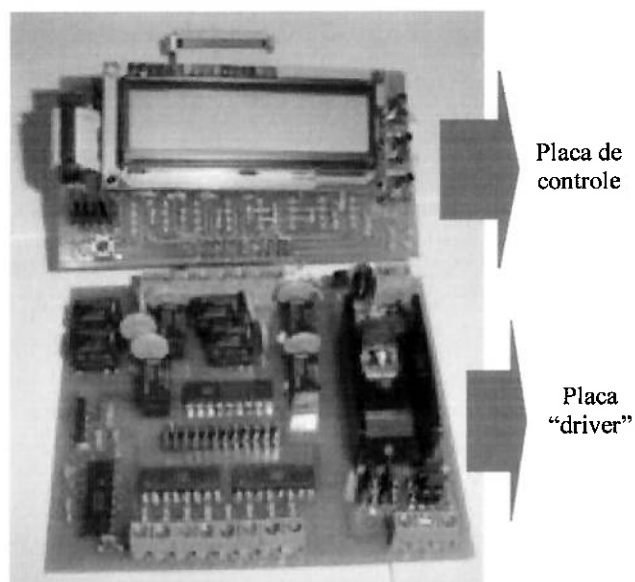


Figura 4.3. Foto com as duas placas eletrônicas do microCLP

A Figura 4.4 mostra detalhes da placa “driver” do microCLP. As regiões em evidência na foto indicam as principais partes que compõem a placa. O regulador de tensão, que fornece a voltagem de 5V DC para o microCLP. As saídas digitais com os relés dispostos na parte superior da foto. As entradas digitais se encontra na parte inferior da figura. O “driver” da comunicação serial se encontra na parte esquerda da foto e na parte direita inferior a entrada analógica.

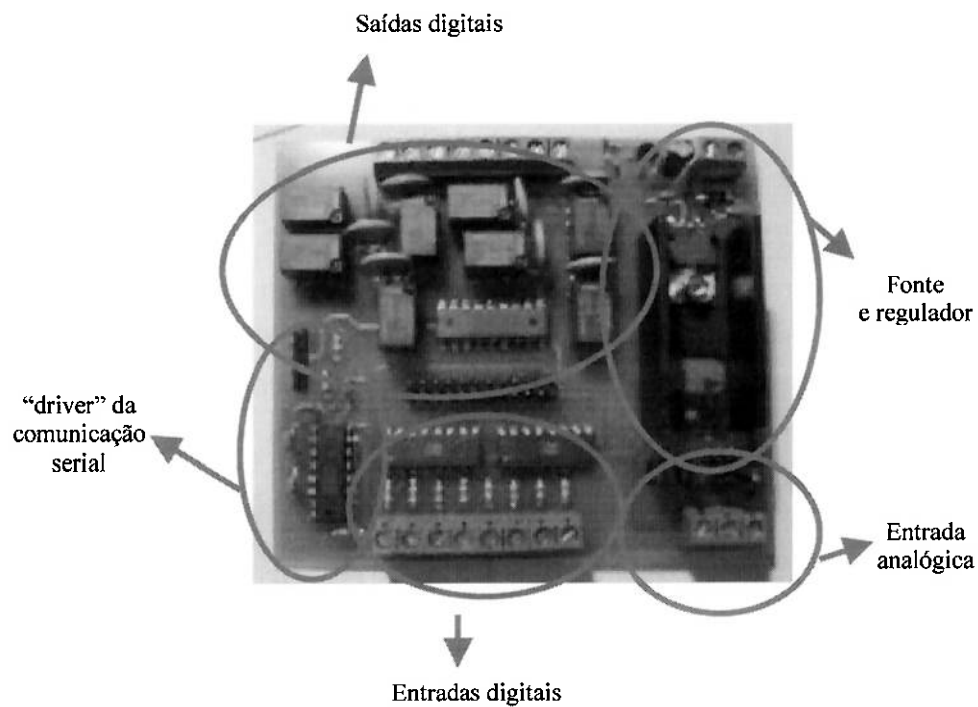


Figura 4.4. Detalhe dos componente da placa driver

*o firmware?
como foi implementado?*

4.2. Edição de Programas de Controle

A figura 4.5 ilustra a tela de trabalho desta interface de programação.

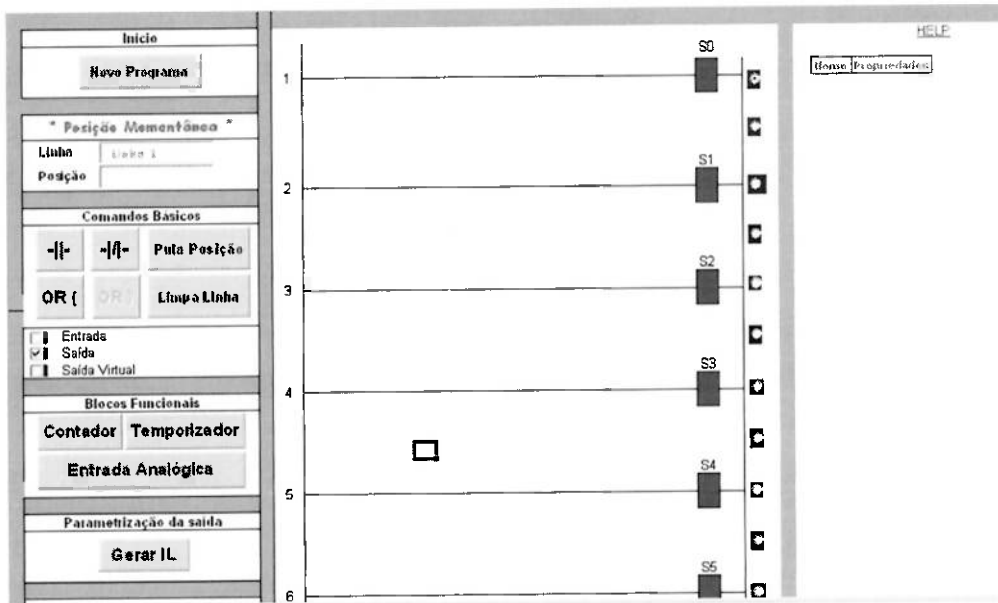


Figura 4.5. Tela de trabalho da interface de programação

Na Figura 4.5 pode-se verificar na parte central o diagrama LD para que o usuário possa acompanhar a lógica que está sendo programada e verificar as variáveis que estão sendo usadas.

O campo do diagrama é composto por linhas horizontais numeradas do lado esquerdo, com um bloco como terminação do lado direito e um campo para selecionar a linha. Essas são as linhas principais de cada expressão que se deseja inserir na lógica do diagrama, nelas são adicionados os contatos e as bobinas.

A composição de cada linha segue o padrão descrito na seção sobre LD, onde os contatos definem a expressão lógica do lado esquerdo e a bobina, que recebe o resultado da lógica, é o último componente da linha no lado direito.

Na coluna a esquerda da tela de trabalho tem-se os componentes que representam as variáveis, isto é, os comandos básicos e os blocos funcionais.

No campo de comandos básicos, tem-se os contatos normalmente abertos, os contatos normalmente fechados e a opção de selecionar a operação lógica para a linha selecionada.

Quanto ao campo dos blocos funcionais, tem-se os blocos descritos nas seções anteriores. Quando selecionados, são introduzidos no diagrama como um tipo de contato e seus parâmetros são configuradas as partes em outro campo do programa.

Na coluna à direita da tela de trabalho tem-se um resumo das variáveis utilizadas em cada linha.

Em resumo, a interface de programação foi implementada considerando os seguintes pontos:

- 11 comandos
- 9 linhas de instruções
- 1 linha adicional para a operação OR para cada linha de instrução
- 5 posições por linha + 1 posição de saída (bobina)
- Campo de controle da posição momentânea da linha sendo editada
- Nome da variável é preenchido automaticamente
- A primeira posição de cada linha válida é deve possuir algum contato válido

1- Descrição dos comandos:

A Figura 4.6 ilustra os comandos da interface de programação.

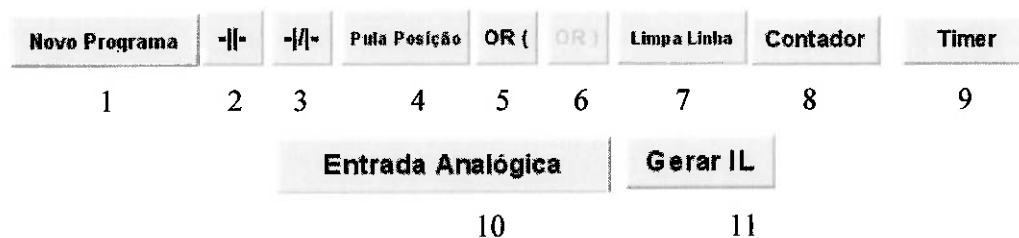
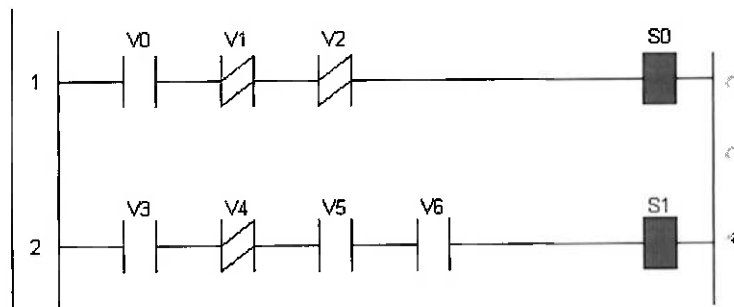


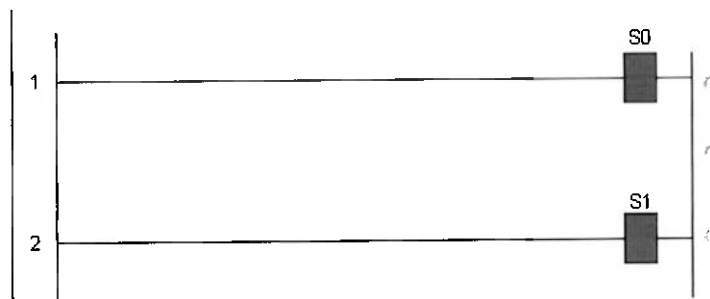
Figura 4.6. Comandos da interface de programação

A. Comando “Novo Programa”

O comando “Novo Programa” permite que o usuário limpe o programa anterior e insira um novo LD (vide Figura 4.7).



(a) Exemplo de um programa de controle em edição



(b) Programa de controle após o acionamento do comando “Novo Programa”

Figura 4.7. Comando “Novo Programa”

B. Comando “Contato Normalmente Aberto”

O comando “Contato Normalmente Aberto” permite que o usuário insira um contato normalmente aberto (Figura 4.8) no programa.

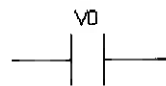


Figura 4.8. Contato Normalmente Aberto

C. Comando “Contato Normalmente Fechado”

O comando “Contato Normalmente Fechado” permite que o usuário insira um contato normalmente fechado (Figura 4.9) no programa.



Figura 4.9. Contato Normalmente Fechado

D. Comando “Pula Posição”

O comando “Pula Posição” permite que o usuário pule uma posição horizontal na linha sendo editada.

E. Comando “Abre OR”

O comando “Abre OR” permite que o usuário crie uma nova linha adicional paralela à linha de trabalho atual (Figura 4.10).

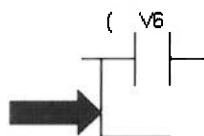


Figura 4.10. Início da linha OR

F. Comando “Fecha OR”

O comando “Fecha OR” permite que o usuário feche a linha adicional paralela à linha de trabalho atual (Figura 4.11).

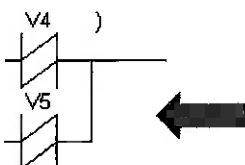
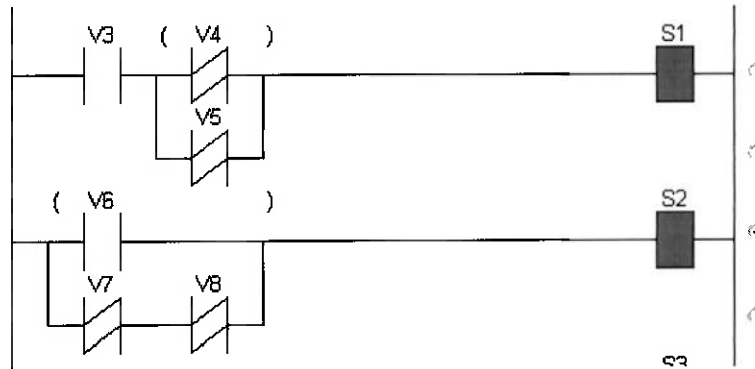


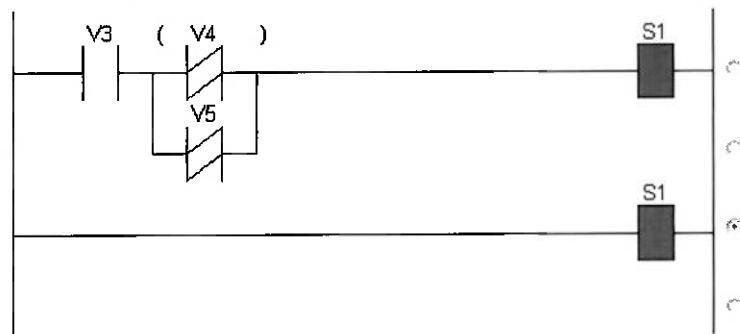
Figura 4.11. Fechamento da linha OR

G. Comando “Limpa Linha”

O comando “Limpa Linha” limpa apenas a linha atual de trabalho.



(a). Exemplo de um programa de controle em edição



(b). Programa de controle após o acionamento do comando “Limpa

Figura 4.12. Linha apagada

H. Comando “Contador”

O comando “Contador” permite que o usuário insira um bloco contador no programa (Figura 4.13).

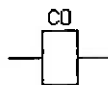


Figura 4.13. Bloco contador

Ao inserir um contador no diagrama, no lado direito da tela de trabalho, aparece uma tabela com o nome da variável e seus parâmetros. Para o contador são 3 parâmetros e inicialmente eles estarão preenchidos com 0 (zero) como valor padrão (Figura 4.14).

Nome	Propriedades		
CO	0	0	0

Figura 4.14. Representação das configurações do bloco contador

Os parâmetros neste caso estão listados na tabela 4.1.

Tabela 4.1. Resumo dos parâmetros do bloco contador

Param.	Valor	Significado
1	0 -255	Número de incrementos a serem contabilizados
2	48	Incremento com a borda de subida do sinal
2	49	Incremento com a borda de descida do sinal
3	48	O bloco fecha o contato caso o número de incrementos seja atingido
3	49	O bloco abre o contato caso o número de incrementos seja atingido

I. Comando “Temporizador”

O comando “Temporizador” permite que o usuário insira um bloco temporizador no programa (Figura 4.15).

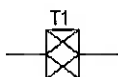


Figura 4.15. Bloco temporizador

Ao inserir um temporizador no diagrama, no lado direito da tela de trabalho, aparece uma tabela com o nome da variável e seus parâmetros. Para o temporizador são 3 parâmetros e inicialmente elas estarão preenchidas com 0 (zero) como valor padrão (Figura 4.16).

Nome	Propriedades		
TO	0	0	0

Figura 4.16. Representação das configurações do bloco temporizador

Os parâmetros neste caso estão listados na tabela 4.2.

Tabela 4.2. Resumo das configurações do bloco temporizador

Parâm	Valor	Significado
1	0 -255	Intervalo de tempo especificado
2	48	Unidade em segundos
2	49	Unidade em décimos
2	50	Unidade em minutos
3	48	O bloco fecha o contato pelo tempo estipulado e depois o abre
3	49	O bloco abre o contato pelo tempo estipulado e depois o fecha
3	50	O bloco espera o tempo estipulado e depois abre o contato
3	51	O bloco espera o tempo estipulado e depois fecha o contato

Comando “Entrada Analógica”

O comando “Entrada Analógica” permite que o usuário insira um bloco entrada analógica no programa (figura 4.17).

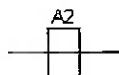


Figura 4.17 Bloco entrada analógica

Ao inserir uma entrada analógica no diagrama, no lado direito da tela de trabalho, aparece uma tabela com o nome da variável e seus parâmetros. Para a entrada analógica são 2 os parâmetros e inicialmente elas estarão preenchidas com 0 (zero) como valor padrão.

Nome	Propriedades		
A0	0	0	0

Figura 4.18 Representação das configurações do bloco entrada analógica

Os parâmetros neste caso estão listados na tabela 4.3.

Tabela 4.3 Resumo das configurações do bloco entrada analógica

Parâm.	Valor	Significado
1	0 -255	Valor esperado na entrada analógica (“set point”)
2	48	Sinal medido cruza o “set point” subindo
2	49	Sinal medido cruza o “set point” descendo
3	48	O bloco fecha o contato caso o valor esperado seja atingido
3	49	O bloco abre o contato caso o valor esperado seja atingido

J. Comando “Gera IL”

O comando “Gera IL” permite que o usuário, após editar o programa de controle, o envio das informações para o microCLP.

2 - Descrição das linhas de instrução do LD:

A linha de instruções é onde o usuário insere os comandos desejados (Figura 4.19).

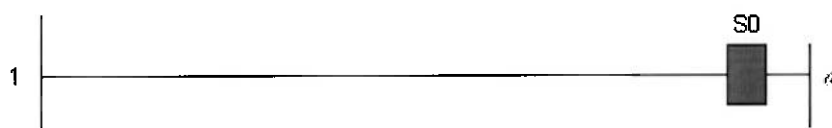



Figura 4.19 Exemplo de linha de instruções


Para editar a linha de instruções, o usuário deve clicar sobre a caixa de opção () presente ao lado direito da linha.

3 - Descrição das linhas adicionais (OR):

A linha adicional paralela a linha de instruções é onde o usuário insere os comandos quando desejar realizar um comando OR (Figura 4.20).



Figura 4.20 Exemplo de uma linha adicional OR

Para editar a linha adicional, o usuário deve clicar sobre a caixa de opção () presente ao lado da linha.

4 – A disposição dos comandos na linha:

Cada uma das linhas (linhas de instruções e linhas adicionais) do diagrama possui 5 posições a serem editadas mais uma posição de saída (bobina).

5 – Descrição do campo de controle da posição momentânea da linha em edição:

O campo de controle da posição momentânea da linha em edição permite que o usuário saiba qual posição da linha está sendo editada no momento (Figura 4.21).



* Posição Momentânea *	
Linha	Or - Linha 1
Posição	Posição 3

Figura 4.21 Indicação da posição e linha em edição

6 – Preenchimento automático dos nomes das variáveis:

Ao editar uma linha de instrução do diagrama com um contato, um temporizador, um contador ou uma entrada analógica, o nome desta variável é preenchido automaticamente, com a seguinte regra indicada na figura 4.22:

- ☐ Entrada
- ☐ Saída
- ☒ Saída Virtual

Figura 4.22 Caixa de seleção para opção de nome das variáveis

6.1.Caso a opção “Entrada” esteja selecionada :

Para que o nome da variável seja uma entrada, basta o usuário clicar na caixa de seleção ilustrada na Figura 4.22, a opção “Entrada”. Assim, o nome de cada variável é preenchido com a letra E.

6.2.Caso a opção “Saída” esteja selecionada:

Para que o nome da variável seja uma saída, basta o usuário clicar na caixa de seleção ilustrada na Figura 4.22, a opção “Saída”. Assim, o nome de cada variável é preenchido com a letra S.

6.3.Caso a opção “Saída Virtual” esteja selecionada:

Para que o nome da variável seja uma saída virtual, basta o usuário clicar na caixa de seleção ilustrada na Figura 4.22, a opção “Saída Virtual”. Assim, o nome de cada variável é preenchido com a letra V.

Caso o usuário queira alterar o nome da variável de algum contato já inserido, basta alterar o nome da que aparece sobre o contato desenhado no diagrama.

7 – Regra de validação de cada linha:

Para que uma linha seja válida e compilada, sua primeira posição deve conter um contato válido (Figura 4.23).

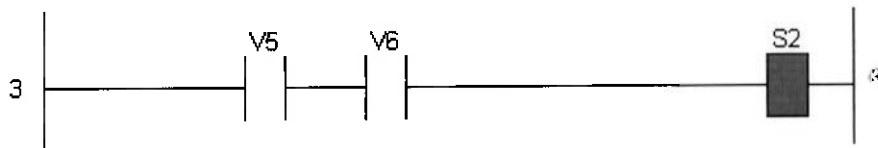
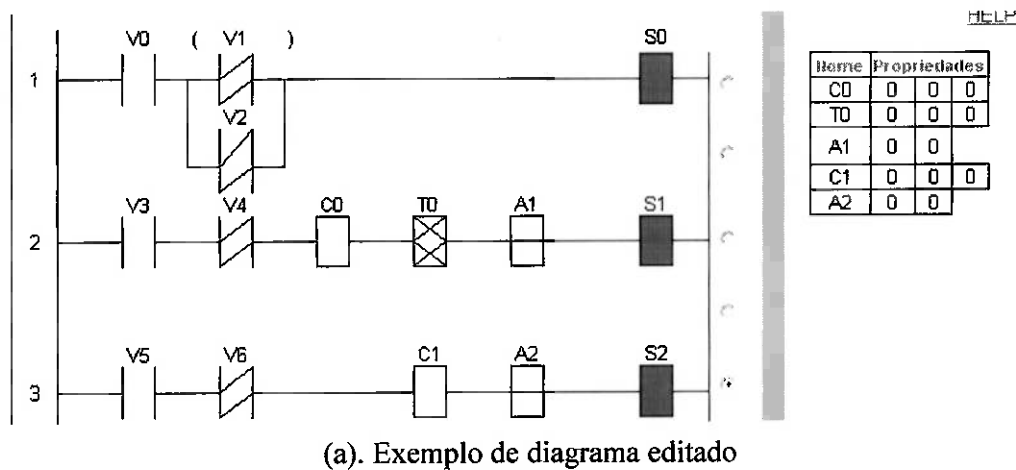


Figura 4.23 exemplo de linha inválida (1ª posição não contém um contato válido)

4.3. Tratamento dos dados do programa de Controle

Com as linhas de instrução devidamente editadas, a interface de programação automaticamente transforma o diagrama LD em códigos de uma planilha auxiliar interna (Tabela de entradas - Figura 4.24)



Entradas do Diagrama de Relés										
3		V0	(V1)					S0
		1	4	0	5					
6			V2							
			0							
9		V3		V4		C0		T0		A1
		1		0		1		1		1
12										
15		V5		V6				C1		A2
		1		0				1		1
18										

(b) Tabela interna com as entradas do diagrama editado

Figura 4.24. Tabela gerada com base nas entradas do LD editado

A “Tabela de Entradas” possui uma regra de preenchimento que é representada na tabela 4.4.

Tabela 4.4. Códigos utilizados na “Tabela de Entradas”

Comando	Valor
Contato Normalmente Aberto	1
Contato Normalmente Fechado	0
Temporizador	1
Contador	1
Entrada Analógica	1
Abre Linha Adicional (OR)	4
Fecha Linha Adicional (OR)	5

Após clicar no comando “Gera IL” (**Gerar IL**), os valores da tabela “Tabela de Entradas” são utilizados para a compilação e geração da lista de instruções (Instruction List), tanto em números hexadecimais quanto em código ASCII.

INSTRUCTION LIST	CÓDIGOS - Decimais					CÓDIGOS ASCII			
1 LD V0	1	116	48	48	48	1 t	0	0	0
2 ANDN(V1	2	132	48	48	48	2 u	0	0	0
3 ORN V2	3	114	48	48	1	3 r	0	0	0
4)	4	117	48	48	48	4 u	0	0	0
5 ST S0	5	115	48	48	1	5 s	0	0	0
6	6	170	170			6 #	#		
7 LD V3	7	49	86	48	48	7 1	V	0	0
8 ANDN V4	8	169	86	49	48	8 @	V	1	0
9 AND C0	9	130	86	50	48	9 ,	V	2	0
10 AND T0	10	58	58	48	48	10 :	:	0	0
11 AND A1	11	50	83	48	48	11 2	S	0	0
12 ST S1	12	170				12 #			
13	13	49	86	51	48	13 1	V	3	0
14 LD V5	14	129	86	52	48	14 □	V	4	0
15 ANDN V6	15	51	67	48	48	15 3	C	0	0
16 AND C1	16	51	84	48	48	16 3	T	0	0
17 AND A2	17	51	65	49	48	17 3	A	1	0
18 ST S2	18	50	83	49	48	18 2	S	1	0
19	19	170				19 #			
20	20	49	86	53	48	20 1	V	5	0
21	21	129	86	54	48	21 □	V	6	0
22	22	51	67	49	48	22 3	C	1	0
23	23	51	65	50	48	23 3	A	2	0
24	24	50	83	50	48	24 2	S	2	0
25	25	170	170			25 #	#		

Figura 4.25. Resultado da compilação do LD

4.4. Estrutura do programa de controle

Variáveis do sistema

Na linguagem IL, as variáveis utilizadas como parâmetro das instruções representam os componentes do sistema, como as entradas e saídas digitais, sendo assim as variáveis são booleanas. Blocos de funções de temporizador, contador e entrada analógica as configurações são variáveis inteiras e não booleanas, mas são configuradas a parte e estão fora do diagrama. No LD do microCLP, as entradas dos blocos são representadas por bobinas.

Bobinas virtuais também estão presentes e podem ser utilizadas como a saída digital.

Na Tabela 4.5 tem-se uma descrição da quantidade das variáveis que podem ser utilizadas na programação do microCLP. Temos oito entradas digitais, oito saídas digitais, oito bobinas virtuais, três blocos possíveis de timer com configuração diferente, cada um deles representando um contato como saída, mas com duas bobinas para ativar a contagem. Três blocos de contador com configurações independentes e três blocos utilizando a leitura da única entrada analógica do sistema, mesmo assim possuem configurações independentes.

Tabela 4.5. Lista de variáveis presentes na programação

Tipo	nomes	explicação
Entradas digitais	E0, E1,..., E7	Contatos equivalentes às entradas
Saídas digitais	S0, S1,..., S7	Bobinas equivalentes às saídas
Bobinas virtuais	V0, V1,..., V7	Bobinas sem equivalente externo
Contato do timer	T0, T1, T2	Contato para à saída do bloco
Contato do contador	C0, C1, C2	Contato para à saída do bloco
Contato da ent. analógica	A0, A1	Contato para à saída do bloco
Bobinas do timer	TEx, TRx, x de 0 à 2	Duas entradas para cada bloco
Bobinas do contador	Cix, CRx, x de 0 à 2	Duas entradas para cada bloco
Bobinas da ent. analógica	AEx, Arx, x de 0 à 1	Duas entrada para cada bloco

Códigos equivalentes para instruções e variáveis

O programa que é carregado no microCLP não precisa conter nomes para as instruções. Como que as instruções já estão em uma lista interna, para economizar o número de bytes que têm que ser enviado e armazenado no microCLP, são definidos códigos equivalentes a cada instrução do IL do microCLP (Tabela 4.7).

Como foi visto na descrição da linguagem IL, existem dois tipos de modificações que podem ser aplicadas as instruções e que modificam a ação das mesmas. Na Tabela 4.6, temos os dois transformadores e o código equivalente na linguagem do microCLP.

O transformador “N” denota “normalmente fechado”, ou seja, ele aparece para representar um contato normalmente fechado. Quando aplicado a instrução ele provoca a inversão do valor da variável, antes de se executar a instrução. O transformador “(“ denota relação entre as instruções do programa. Quando uma instrução tem uma relação de dependência com relação ao resultado da operação posterior, o transformador é utilizado.

Tabela 4.6. Lista de códigos para os transformadores utilizados com as instruções

Transformadores	Código (um byte em decimal)
N	78
(40

Tabela 4.7. Lista de códigos equivalentes para instruções

Valor em IL (Comandos)	Valor em Decimal
LD	49
ST	50
AND	51
OR	52
ADD	53
SUB	54
GE	55
EQ	56
LE	57
)	58
CALL	59
RETURN	60
LDN	127
AND(91
ANDN	129
ANDN(169
OR(92
ORN	130
ORN(170

A Tabela 4.8 mostra todas as variáveis do sistema, pode-se notar que os códigos são formados por um byte que define a natureza da variável, ou seja, se é contato, se é bobina, se é bobina virtual ou se é um bloco funcional, o outro byte é o número específico de cada variável.. Cada combinação desses dois bytes representa uma variável.

Tabela 4.8. Lista de códigos para as variáveis do sistema

Variáveis	Código (dois ou três bytes)
E0 ... E7	69; 48 – 55
S0 ... S7	83; 48 – 55
V0 ... V7	86; 48 – 55
T0 ... T2	84; 48 – 50
C0 ... C2	67; 48 – 50
A0 ... A2	65; 48 – 49
TE0 ... TE2	84; 69; 48 – 50
TR0 ... TR2	84; 82; 48 – 50
CI0 ... CI2	67; 73; 48 – 50
CR0 ... CR2	67; 82; 48 – 50
AE0 ... AE2	65; 69; 48 – 49

Cabeçalho

O cabeçalho do programa de controle contém as configurações de cada bloco funcional que é utilizado no programa em IL, caso nenhum bloco esteja presente no programa, nenhuma configuração aparecerá no cabeçalho.

O cabeçalho é dividido em blocos de quatro bytes que definem as configurações de cada bloco, como descrito no item de IL para microCLP

Um exemplo de cabeçalho está na Tabela 4.9.

Tabela 4.9. Cabeçalho do programa de controle

Bytes de configuração			
1	2	3	4
116	148	48	48
132	98	49	48
173	173	173	173

Nome	Propriedades		
C0	100	0	0
T0	50	1	0

Figura 4.27. Configuração equivalente a Tabela 4.9

Tabela 19. Regras para compilação das configurações para o cabeçalho

Valor em IL (Variáveis)	Valor em Decimal
T0	132
T1	133
T2	134
C0	116
C1	117
C2	118
A0	113
A1	114

IL

As instruções, da mesma forma que o cabeçalho são convertidas para blocos de 4 bytes, onde o primeiro representa o código da instrução e os dois ou três seguintes representam a variável utilizada com a instrução. Na Tabela 4.10, pode-se analisar um exemplo de código.

Tabela 4.10. Exemplo de IL convertido para os códigos do microCLP

	Cód. IL	Variável	Bytes equivalentes as instruções			
			1	2	3	4
1	LD	E0	49	69	48	48
2	AND	E1	51	69	49	48
3	ST	S0	50	83	48	48
4			173	173	173	173

4.5. Comunicação com o microCLP

Uma vez gerada a tabela em código em ASCII, a interface de programação gera um arquivo de saída no formato texto (extensão “SAIDA.txt”). Este arquivo possui todos os dados do diagrama construído em código ASCII e é utilizado no carregamento do programa de controle no microCLP.

Para enviar o programa de controle ao microCLP, a interface de programação executa o programa “microCLP” (microCLP.exe) que monitora, loga, e envia os dados do arquivo texto para o módulo de execução. A figura 4.28 ilustra a tela do programa “microCLP”.

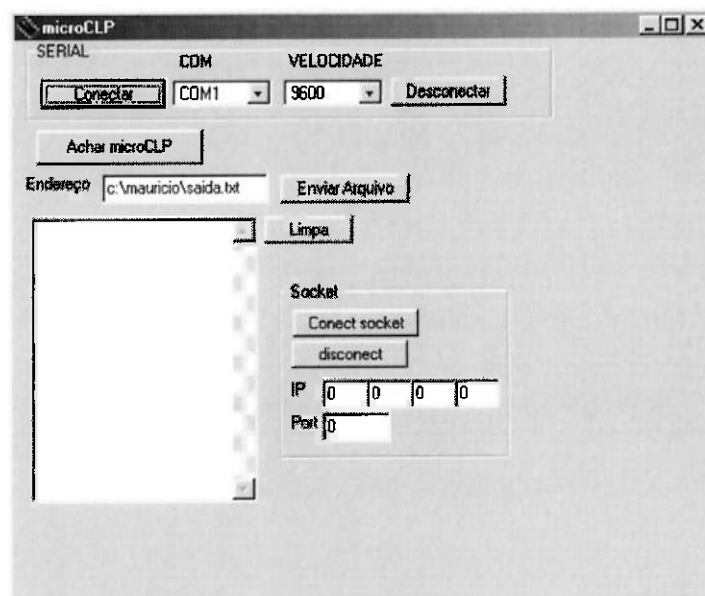


Figura 4.28 Tela principal do programa “microCLP”

A seguir o usuário deve seguir os seguintes passos:

1. Clicar no comando “Conectar”

No campo de log, o programa mostra o status da conexão com a porta COM do microcomputador que está executando a interface de programação (Figura 4.29).

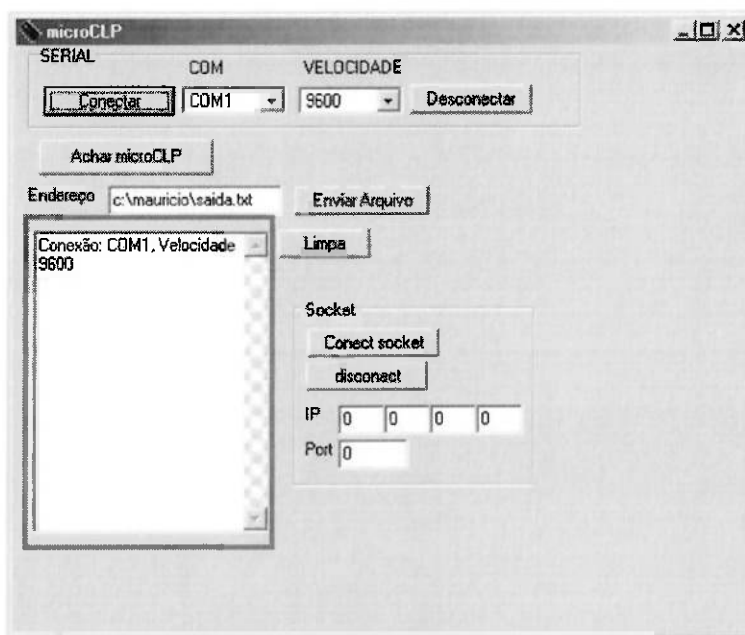


Figura 4.29 Indicação da abertura da porta COM

2. Clicar no comando “Achar microCLP”

O comando verifica se o microCLP está conectado à porta COM do microcomputador e se seu status está como “Comunicação”. O programa gera a seguinte mensagem no campo log: “MICRO-CLP IS ON” (Figura 4.30).

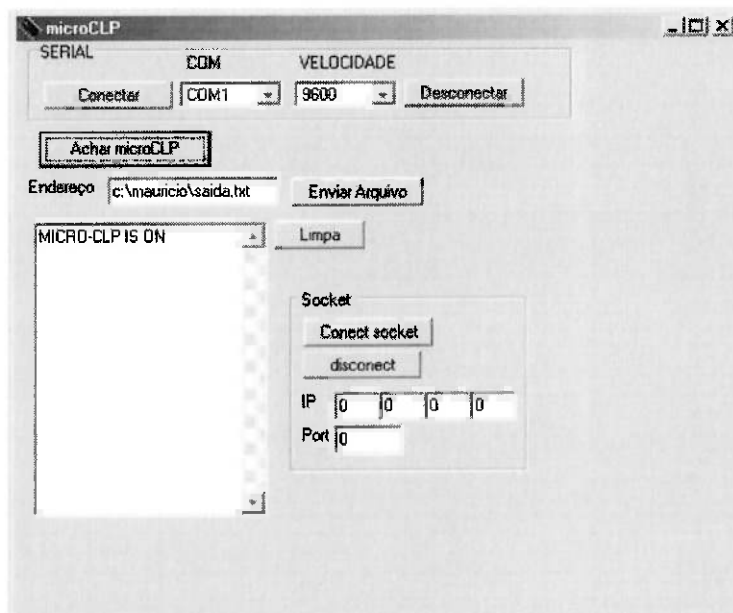


Figura 4.30 Programa confirmando a comunicação com o microCLP

3. Especificar o endereço do diretório aonde o arquivo de saída está localizado. O endereço padrão utilizado pela interface de programação já está gravado (c:\microCLP\saida.txt).
4. Clicar no comando “Enviar Arquivo”

A princípio, o programa verifica se o arquivo “Saida.txt” existe no endereço do diretório especificado acima. Caso não, o sistema gera a seguinte mensagem de aviso: “FILE NOT FOUND”

Caso o arquivo esteja no local indicado, o sistema gera a seguinte mensagem no campo log: “Arquivo Saida.txt OK!!!”

Confirmada a existência do arquivo, o programa passa a monitorar o envio dos dados e sua gravação no microCLP. Obtendo êxito na gravação, o módulo gera a seguinte mensagem no campo log: “PROGRAMA OK” (Figura 4.31).

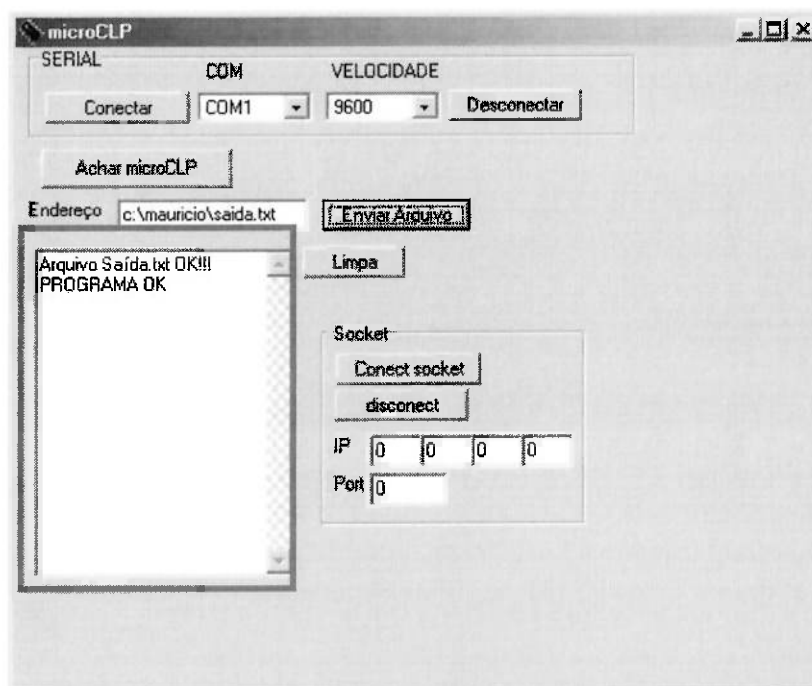


Figura 4.31. Mensagem de programa carregado no microCLP

Assim, executados os passos acima listados, o programa de controle em LD editado na interface de programação está carregado no microCLP e pronto para execução.

Controle do envio de dados

A comunicação entre o microCLP e a interface é feita por uma interface de comunicação serial assíncrona, ou seja, é uma comunicação feita a três fios. Um sendo o transmissor, outro o receptor e outro a referência com o sinal de “terra”.

Esse tipo de comunicação não apresenta sinal de controle de fluxo com CTS, RTS e etc, normalmente presentes na comunicação síncrona.

Para fazer o controle do fluxo nesse caso, é necessária a comunicação baseada em mensagens onde a mensagem inclui alguns bytes de controle. No caso do microCLP, utilizam-se dois bytes de controle.

Cada mensagem enviada ao microCLP pela interface, tem como o primeiro byte um código que chama uma função. Os códigos estão listados na Tabela 4.11.

Tabela 4.11. Relação entre os códigos do primeiro byte e as funções do microCLP

Byte(decimal)	Função do microCLP
1	Chama a função “keep alive” para verificar se o microCLP está conectado a serial e se está configurado na opção de comunicação serial
2	Chama a função que escreve um bloco de quatro bytes de uma instrução na memória do microCLP
4	Chama a função que finaliza a programação do programa e verifica se não ocorreram erros

Um último byte é utilizado com a mensagem de programação do código, para dar mais segurança à mensagem de programação de instruções (Código 2). O último byte é um byte de “check sum” dos cinco bytes anteriores. A mensagem recebida pelo microCLP pode ser verificada quanto a sua integridade antes de ser gravada na memória do microCLP.

Tabela 4.12. Estrutura da mensagem de programação dos códigos no microCLP

Estrutura das mensagens de programação					
1	2	3	4	5	6
Código de comando	02	48	69	48	Check sum

5. TESTES DO PROTÓTIPO DO MICROCLP

5.1. Testes de hardware

Para os testes de hardware foi desenvolvida uma placa de acrílico contendo botões conectados as entradas digitais e LEDs conectados as saídas digitais.

A placa ainda contém um trilho DIM para fixação do microCLP, um botão liga/desliga geral, um potenciômetro de 10k conectado a entrada analógica e uma fonte de 24 Volts que alimenta todo o sistema.

A Figura 5.1 ilustra um visão geral da placa de testes com o microCLP instalado.

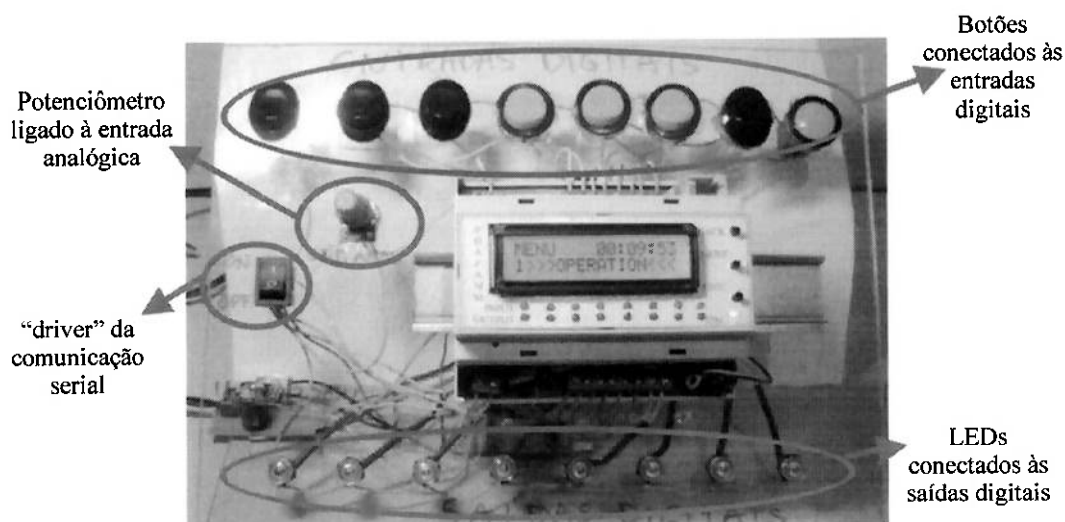


Figura 5.1. Foto da placa de testes dos recursos do microCLP

5.2. Teste da Interface de programação e do firmware

O exemplo na Figura 5.2 ilustra as principais características em relação ao programa de controle e os recursos para sua edição na interface de programação e execução no microCLP.

LD

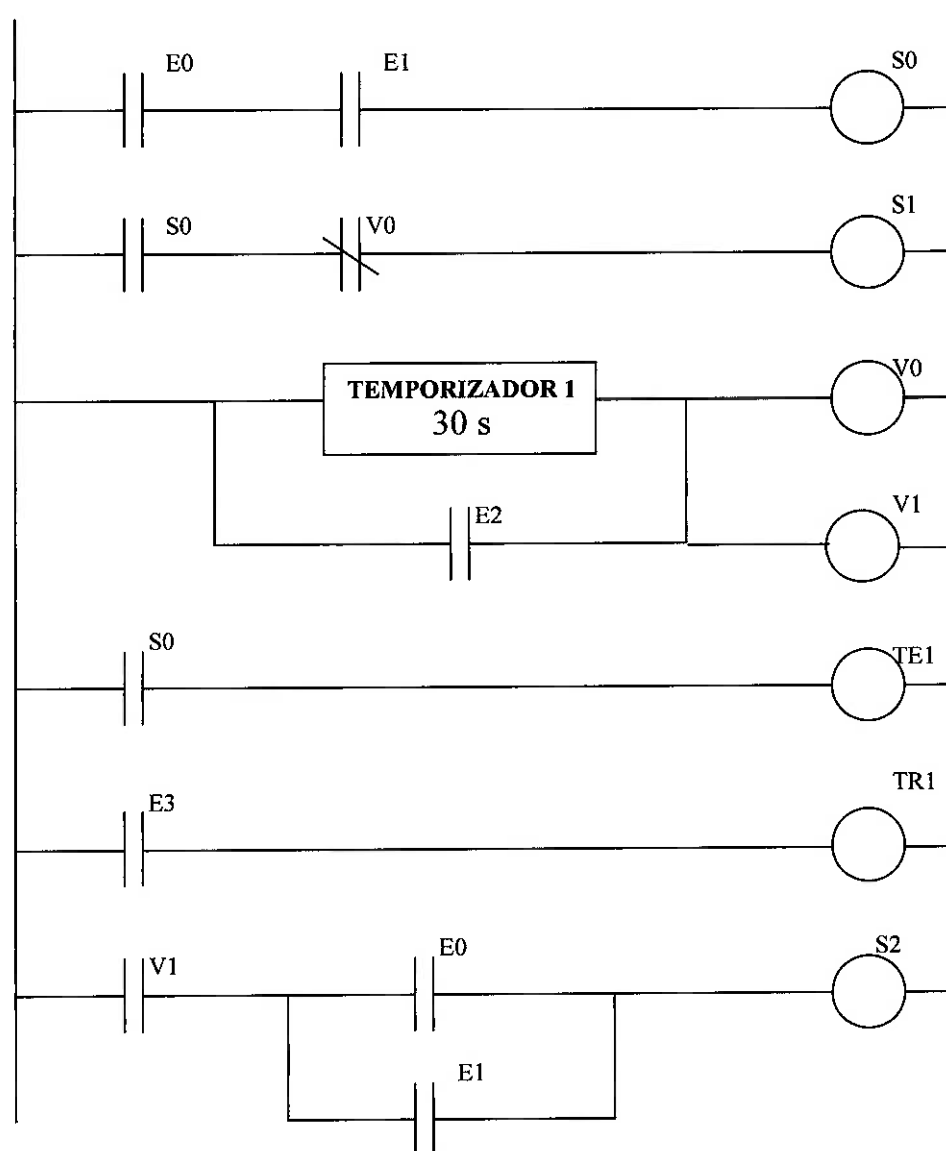


Figura 5.2. Exemplo de um programa de controle em LD

IL

O programa de controle em LD da Figura 5.1 deve ser convertido pela interface de programação em IL da Tabela 5.1

Tabela 5.1. Programa em IL do exemplo da Figura 5.1

	Código	Variável
1	LoaD	E0
2	AND	E1
3	STore	S0
4		
5	LoaD	S0
6	ANDN	V0
7	STore	S1
8		
9	LoaD	T1
10	OR	E2
11	Store	V0
12	STore	V1
13		
14	LoaD	S0
15	STore	TE1
16		
17	LoaD	E3
18	STore	TR1
19		
20	LoaD	V1
21	AND(E0
22	OR	E1
23)	
24	STore	S2
25		

Equacionamento das instruções

As equações são representações de cada linha horizontal que liga as linhas de energia do programa de controle em LD.

$$S0 = (E0 * E1) \quad \text{Eq. (5.1)}$$

$$S1 = (S0 * \overline{V0}) \quad \text{Eq. (5.2)}$$

$$V0 = V1 = (T1 + E2) \quad \text{Eq. (5.3)}$$

$$TE1 = (S0) \quad \text{Eq. (5.4)}$$

$$TR1 = (E3) \quad \text{Eq. (5.5)}$$

$$S2 = (V1 * (E0 + E1)) \quad \text{Eq. (5.6)}$$

O programa gerado pela interface de programação que é enviado para o microCLP é equivalente ao programa em IL, apenas é adaptado para ocupar o menor número de bytes e ter uma estrutura que seja mais fácil de ser interpretada pelo microCLP. A Tabela 5.2 apresenta a lista de bytes que é carregada e executada no microCLP.

Equacionamento das instruções

As equações são representações de cada linha horizontal que liga as linhas de energia do programa de controle em LD.

$$S0 = (E0 * E1) \quad \text{Eq. (5.1)}$$

$$S1 = (S0 * \overline{V0}) \quad \text{Eq. (5.2)}$$

$$V0 = V1 = (T1 + E2) \quad \text{Eq. (5.3)}$$

$$TE1 = (S0) \quad \text{Eq. (5.4)}$$

$$TR1 = (E3) \quad \text{Eq. (5.5)}$$

$$S2 = (V1 * (E0 + E1)) \quad \text{Eq. (5.6)}$$

O programa gerado pela interface de programação que é enviado para o microCLP é equivalente ao programa em IL, apenas é adaptado para ocupar o menor número de bytes e ter uma estrutura que seja mais fácil de ser interpretada pelo microCLP. A Tabela 5.2 apresenta a lista de bytes que é carregada e executada no microCLP.

Tabela 5.2. Lista de bytes equivalente ao programa exemplo da Tabela 5.1

	Cód. IL	Variável	Bytes equivalentes as instruções				
			1 ^o	2 ^o	3 ^o	4 ^o	
			133	30	48	48	cabeçalho
			173	173	173	173	
1	LD <i>load</i>	E0	49	69	48	48	
2	AND	E1	51	69	49	48	
3	ST <i>store</i>	S0	50	83	48	48	
4			173	173	173	173	
5	LD	S0	49	73	48	48	
6	ANDN	V0	129	86	48	48	
7	ST	S1	50	73	49	48	
8			173	173	173	173	
9	LD	T1	49	84	49	48	
10	OR	E2	51	69	50	48	
11	ST	V0	50	86	48	48	
12			173	173	173	173	
13	LD	S0	49	73	48	48	
14	ST	TE1	50	84	69	49	
15			173	173	173	173	
16	LD	E3	49	69	51	48	
17	ST	TR1	50	84	82	49	
18			173	173	173	173	
19	LD	V1	49	86	49	48	
20	AND(E0	169	69	48	48	
21	OR	E1	52	69	49	48	
22)		58	48	48	48	
23	ST	S2	50	73	50	48	
24			173	173	173	173	

Na Tabela 5.2 tem-se o código IL do lado esquerdo e os bytes equivalentes do lado direito. Pode-se notar quatro colunas de bytes, onde dedica-se quatro bytes no máximo, para cada linha de IL. Dessas quatro colunas, a primeira é equivalente a instrução do IL, Load, Store, AND, etc.

Nota-se também que quando a instrução contém um modificador o valor do byte equivalente é modificado. No caso de modificadores o valor do código do modificador e o valor do código da instrução, apresentados nas tabelas do item 4.4, são somados, obtendo-se um novo valor e sem ter que acrescentar mais um byte para diferenciar o código.

Os bytes 2, 3 e 4 são utilizados para a variável utilizada com a instrução. São sempre utilizados três bytes, por conta de variáveis como TR1, CI1, etc. As variáveis que contém apenas dois caracteres ou as instruções que não contém variáveis é utilizado um byte 0x00 para completar o espaço de três bytes.

Entre as expressões são utilizados os bytes 0xAA e no início e no final do programa temos quatro bytes 0xAA.

O programa de controle em LD neste exemplo, convertido em IL gastou 96 bytes. Mas esse não é o tamanho total do programa já que o código deve conter também a configuração do bloco funcional temporizador que é adicionado no início da lista de bytes e contém também quatro bytes, o primeiro byte define o bloco utilizado. No exemplo temos o Temporizador T, representado pelo valor 0x55 que é o valor (equivalente em código ASCII) de T e 1 somados. Os outros três bytes representam as propriedades do bloco. O byte dois é equivalente a unidade segundos, o byte três é equivalente ao valor de 30 segundos e o byte quatro é equivalente ao tipo de saída (4) descrito no item 3.2.1.

o que
foi testado ??

6. CONCLUSÕES

O objetivo de continuar o trabalho sobre esse tema e melhorar o funcionamento do protótipo visou aprofundar o estudo sobre essa área e aplicar isso no projeto de um dispositivo que realmente atue como os semelhantes encontrados no mercado.

Para esse projeto de formatura, pudemos dividir os assuntos que envolviam o projeto em duas partes, uma que tratava de gerar uma interface de programação que permitisse que o usuário final pudesse utilizar linguagens já padronizadas para escrever o programa que desejasse executar no microCLP, a outra parte tratou da elaboração do hardware que implementasse todas as interfaces necessárias ao projeto (I/O, comunicação, usuário) e também um algoritmo de controle que fizesse com que o microCLP funcionasse como um CLP convencional.

Quanto à interface de programação, implementou-se uma interface que é de fácil manipulação e que oferecesse recursos que qualquer usuário acostumado com Diagrama de Relés, pode reconhecer. A transformação do programa de controle de LD para os códigos do microCLP é transparente para o usuário, permitindo que ele trabalhe apenas com a interface gráfica.

Na interface de programação, o resultado obtido foi o esperado. Dentre todas as especificações funcionais impostas para seu desenvolvimento, todos os recursos planejados foram englobados. Assim, as operações de ~~a~~ inclusão de contatos, blocos funcionais ou operações OR são atendidas. A interface construída possui a limitação de número de linhas e posições. Essa limitação deve-se ao fato do programa ser um protótipo.

Quanto à parte de hardware e algoritmo de controle, o resultado foi um microCLP que opera com características elétrica para aplicações industriais (24VDC). Com um “clock” de 40MHz, o microCLP permite que o atraso na atualização das variáveis do sistema seja minimizado, por conta do ciclo de varredura mais rápido. O algoritmo de controle também atingiu o objetivo de operar conforme o programa de controle em LD, que é editado pelo usuário. Além disso, o algoritmo englobou as funcionalidades dos blocos temporizador, contador e entrada analógica.

Englobando essas duas tarefas temos todo um estudo sobre a elaboração de projetos de engenharia envolvendo normas e convenções, que permitiu que o produto

final pudesse ser compatível com aquilo que é comumente utilizado pelo mercado e que permita que ele possa ser integrado aos demais componentes de um sistema de automação.

Os resultados servem como base para aprofundar o estudo na norma e buscar em outros trabalhos aplicar a lógica de multitarefa descrita pela norma.

Como visão geral, o projeto foi uma grande oportunidade para os alunos elaborarem o projeto de um dispositivo aplicado em automação, que engloba fundamentos aprendidos em diversas matérias do curso, como eletrônica digital, analógica, sistemas pneumáticos, controle. Também permitiu vivenciar a tarefa de conduzir um projeto de um sistema de forma que ele pudesse ser entendido e aplicado por outras pessoas, assim o projeto pode vir a evoluir de um protótipo para ser produto.

não é o
objetivo?
!?

7. REFERÊNCIAS

- [1] Caropreso, Rodrigo T., “IEC-61131-3, Vantagens e desvantagens da programação de CLPs”, Mecatrônica Atual, número 15, São Paulo, SP,2004.
- [2] Carvalho, Paulo C., “Controlador Lógico Programável”, Mecatrônica Atual, número 2, São Paulo, SP, 2002.
- [3] Hughes, T A, ”Programmable Controller”, Instrument Society of America, Research Triangle Park, NC,1989
- [4] International Electrotechnical Commission (IEC), “IEC61131-3, International Standard”, resumo, Geneva, Switzerland, 2003
- [5] Marchesan, Ricardo B., “Microautomação, Afinal, o que é isso?”, Mecatrônica Atual, número 16, São Paulo, SP, 2004.
- [6] Microchip Technology, PIC18FXX2 DataSheet family, Chandler, AZ,2001
- [7] Miyagi, P. E. , “Controle Programável: fundamentos do controle de sistemas de eventos discretos”, Edgard Blucher, São Paulo, SP,1996
- [8] Świder J., Wszolek G., Carvalho W, “Controller based on microcontroller designed to execute the logic control of pneumatic systems”, Proceedings of the 12th International Scientific Conference, AMME 2003, Zaczopane, Polônia, pp. 959-964, 2003
- [9] NELLI SILVA, Emílio C, Apostila de pneumática, Departamento de Engenharia Mecatrônica e Sistemas Mecânicos, São Paulo 2002.
- [10] ŚWIDER J., WSZOLEK G., CARVALHO W.: Example of a system prepared to be controlled by the controller based on microcontroller, 12 International Scientific Conference Achievements in Mechanical & Materials Engineering, Gliwice-Zaczopane 2003, Poland

MICROCLP

Manual do usuário

Versão 1.0

TÍTULO DO DOCUMENTO: Manual do usuário - Interface de Controle		AUTOR: Wagner de Oliveira Carvalho
ARQUIVO : Manual do microCLP.doc	Ferramenta utilizada : MS Word 97 / 2000	DATE: 11/12/2004

--	--

Índice

INTRODUÇÃO	3
INSTALAÇÃO E CONEXÕES	6
MENU INTERNO.....	8
RECURSOS	13
PROGRAMAÇÃO DO MICROCLP	14

--	--

INTRODUÇÃO

O propósito deste documento é apresentar os recursos e o funcionamento do microCLP

O que é o microCLP?

O microCLP é um CLP de baixo custo voltado para sistemas de automação de pequeno porte. É composto por interface de I/O, interface comunicação e interface de usuário todas integradas em uma mesma estrutura física, compacta e pronta para ser aplicada para sistemas de automação.

O microCLP dispõe ainda de uma interface de programação que complementa o produto e permite que o usuário possa programa-lo utilizando a linguagem LD, já largamente difundida na área industrial.

Aplicações

O microCLP oferece recursos e possui características que permitem que ele seja aplicado em diversas áreas, incluindo automação predial, residencial ou industrial. Segue abaixo alguns exemplos de aplicação.

- Equipamentos de transporte
 - Controle de esteiras transportadoras
 - Controle de plataformas de elevação
 - Controle de elevadores
 - Controle de silos
 - Controle de alimentadores automáticos

- Instalação elétrica e automação de edifícios
 - Controle de iluminação (exterior e interior)
 - Controle de portas e portões
 - Controle de persianas e toldos
 - Controle de sistemas de irrigação

--	--

➤ Aquecimento/ ventilação/ condicionamento

- Controle de parâmetros de consumo energético
- Controle de aquecedores
- Controle de instalações frigoríficas
- Controle de sistemas de ventilação
- Sistemas de ar condicionado

➤ Máquinas

- Controle de motores, bombas e válvulas
- Controle de compressores de ar
- Controle de sistemas de exaustão e filtragem
- Controle de estações de tratamento de água
- Controle de serras e plainas
- Controle de equipamentos químicos e purificação

➤ Sistemas de monitoramento operacional

- Controle de acesso
- Supervisão de controles de veículos
- Controle de sistemas de alarme
- Monitoração de limites de variáveis
- Controle de sistemas de semáforos
- Controle de transporte de bagagem

➤ Soluções especiais

- Controle de sistemas de energia solar
- Controle aplicado em embarcações
- Controle de painéis indicadores e sinalização de tráfego

Características básicas

- Linguagem de programação em Diagrama de Relés
- Oito entradas digitais opto-acopladas
- Oito saídas digitais à Relé
- Uma entrada analógica
- Interface de usuário contendo um display e botões para manipular as funções do microCLP
- Interface de comunicação serial
- Voltagem de operação em 20V e 30V DC
- Instalação em trilho DIN
- LEDs de status das entradas e saídas

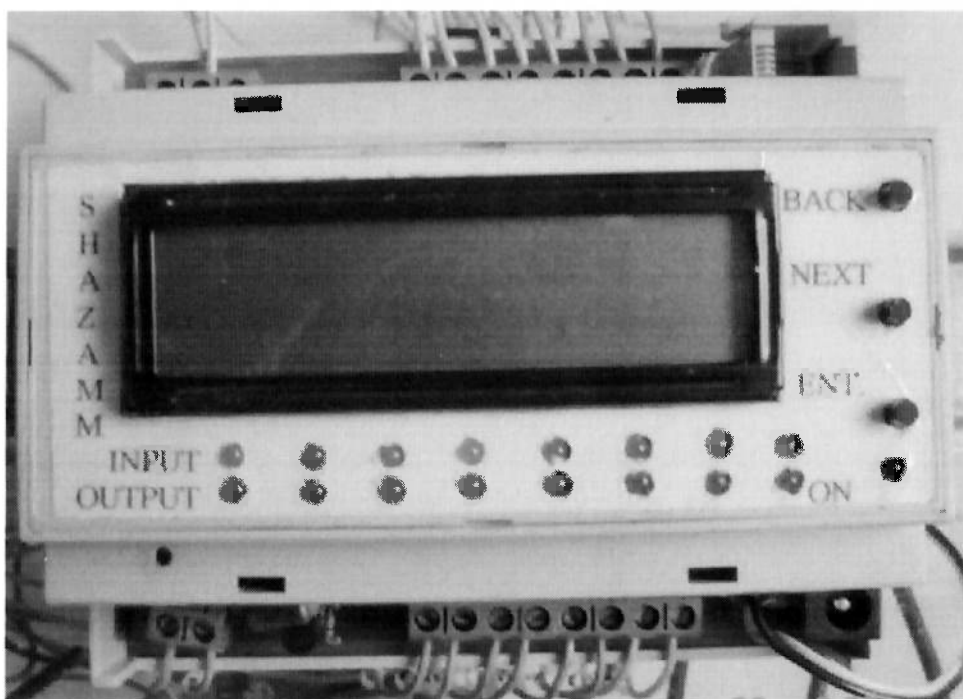


Figura 1. Visão frontal do microCLP

INSTALAÇÃO E CONEXÕES

A instalação do microCLP deve ser feita em trilho ^{DIN} em local protegido, de preferência em caixas elétricas, isoladas e protegidas de umidade, gases e pó.

As entradas e saídas possuem bornes que permitem que os fios sejam fixos com o auxílio de um chave de fenda.

Abaixo seguem fotos das conexões externas.

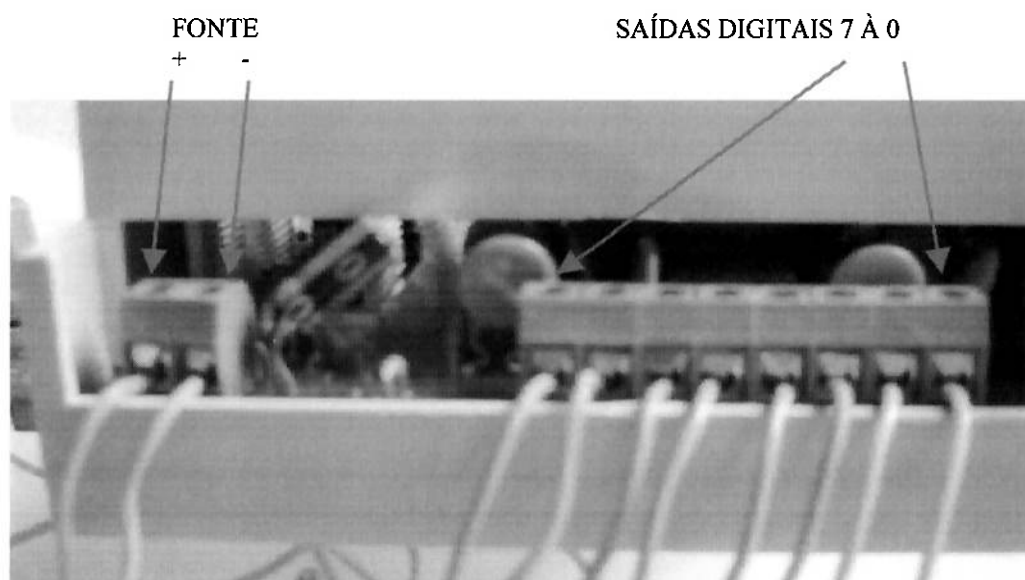


Figura 2. Conexão da fonte e das saídas digitais

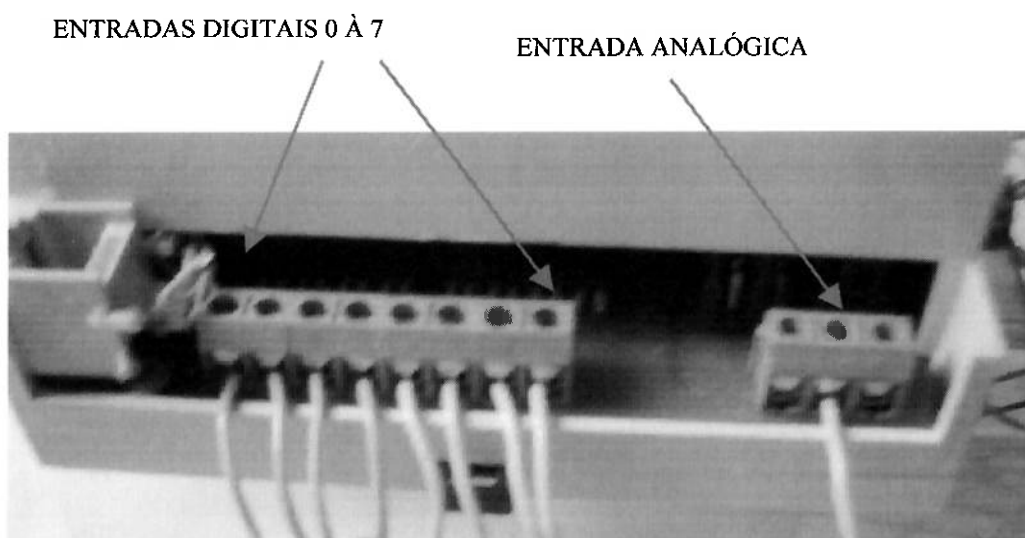


Figura 3. Conexão da entradas digitais e da entrada analógica

--	--

Na Figura 3, pode-se notar que a entrada analógica possui três conexões, a do meio é a entrada do sinal e os dois laterais são conexões diretas da fonte do microCLP, que são uma opção no caso do sensor analógico precisar ser alimentado.

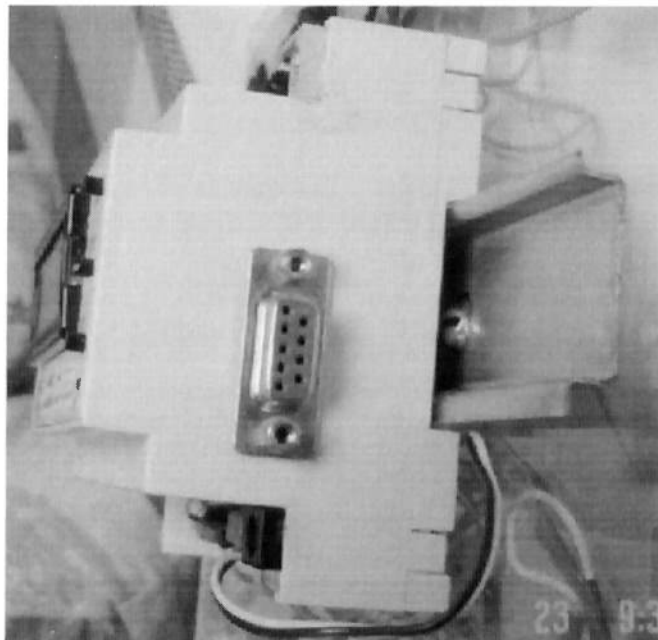


Figura 4. Conexão com a comunicação serial do microCLP

Na Figura 4, tem-se a foto do conector DB-9 fêmea, utilizado para a conexão para a comunicação serial, e também a fixação da caixa no trilho DIN.

MENU INTERNO

Pelo menu apresentado na interface de usuário na parte superior do microCLP não é possível programá-lo, mas é possível fazer testes com os recursos e habilitar a comunicação com a interface de programação.

A interface de usuário é composta por um display LCD 16x2 linhas, onde são mostradas mensagens de acordo com a função desejada. Existem três botões que permitem navegar pelo menu, são eles a tecla “ENTER”, a tecla “NEXT” e a tecla “BACK”.

A interface possui ainda um LED para indicar energização do microCLP, também possui oito LEDs que indicam o status das entradas digitais e também oito LEDs que indicam o status das saídas digitais.



Figura 5. Tela inicial do menu

A Figura 6 mostra outra opção do menu, que é de testar os recursos do microCLP



Figura 6. Opção de testar os recursos do microCLP

--	--

A figura 7 mostra a opção de estabelecer a comunicação serial entre o microCLP e a interface de programação.



Figura 7. Opção de comunicação serial do menu principal

A partir da opção de teste de recursos podem ser escolhidas três opções indicadas abaixo pelas Figuras 8, 9, 10, 11, 12, 13.



Figura 8. Opção de testar se as entradas digitais estão sendo identificadas



Figura 9. Teste das entradas



Figura 10. Opção para testar as saídas digitais

--	--

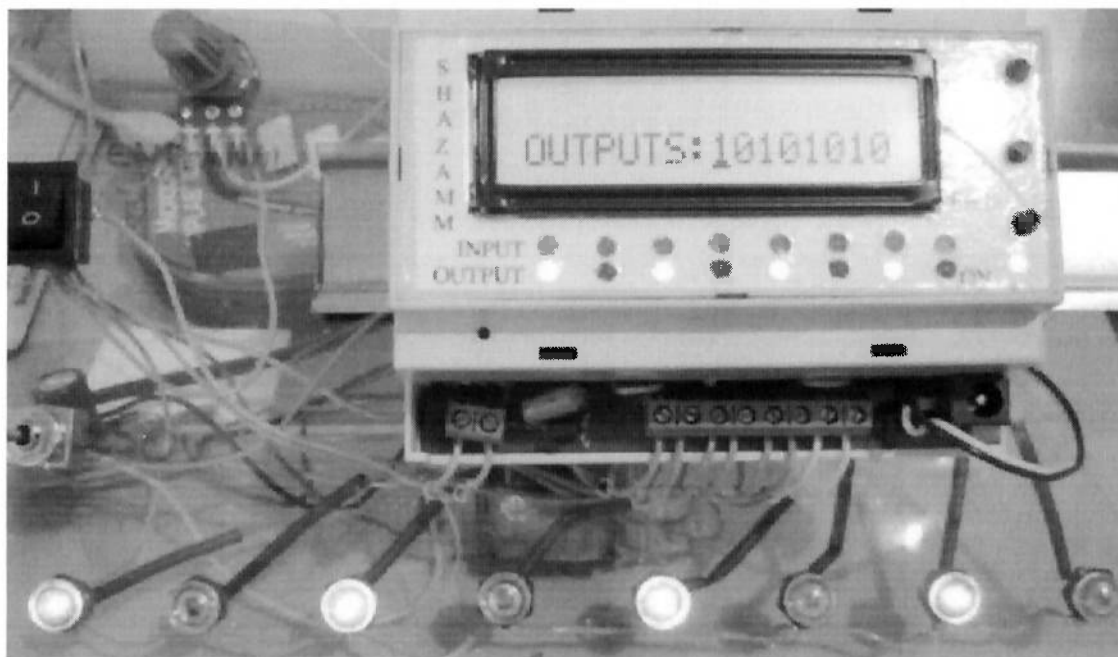


Figura 11. Teste das saídas digitais

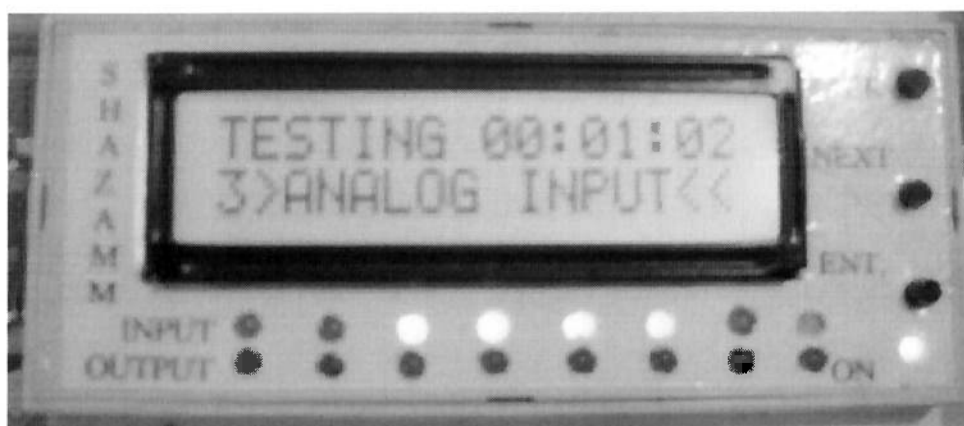


Figura 12. Opção para testar a entrada analógica



Figura 13. Teste da entrada analógica

As outras opções do nível principal do menu são a comunicação serial e a operação do microCLP. Abaixo estão ilustradas as duas opções



Figura 14. Estabelecimento da comunicação serial



Figura 15. Execução do programa de controle

--	--

RECURSOS

Entrada Digital

As entradas digitais do microCLP suportam um nível de tensão de até 30V DC. Para a tensão de 24V DC consome uma corrente de 5mA. As entradas são opto-acopladas o que protege o microCLP de eventuais surtos ou curtos-circuitos que venham a ocorrer.

Saída Digitais

As saídas digitais são à relé, cada relé pode fornecer 800mA de corrente e são protegidos por fusíveis do tipo PTC.

Como o acionamento é por relé, um cuidado especial deve ser dado a frequência de acionamento que não deve passar dos 10Hz para evitar desgaste excessivo dos mesmos.

Entrada Analógica

A entrada analógica está preparada para processar um sinal que varia de 0V à 10V, mas pode também ser adaptada para sinal de corrente de 4mA à 20mA.

A frequência de amostragem é fixa em 1kHz o que exige um certo cuidado com relação a impedância do circuito do sensor. Não deve ser maior do que 10k Ω .

PROGRAMAÇÃO DO MICROCLP

A edição do programa de controle que será utilizado junto com o microCLP deve ser feita na interface de programação que acompanha o microCLP e que deve ser utilizada em um microcomputador que tenha interface serial.

A interface permite que o programa seja elaborado em ~~Diagrama de Relés~~, oferecendo para o usuário a possibilidade de programar o microCLP apenas por essa interface gráfica.

Todos os recursos do microCLP pode ser acessados por meio de botões na interface. Mas detalhes do programa de interface podem ser obtidos no Manual da Interface de Programação.

Depois de ter editado o programa de controle, o usuário acessa um outro programa chamado “microclp.exe”, que irá estabelecer a comunicação serial e irá passar os dados do programa de controle para o microCLP. Segue a sequência de procedimentos

1. Clicar no botão “Conectar”



Figura 16. Botão do programa de envio de dados

No campo de log, o programa mostra o status da conexão com a porta COM do computador.

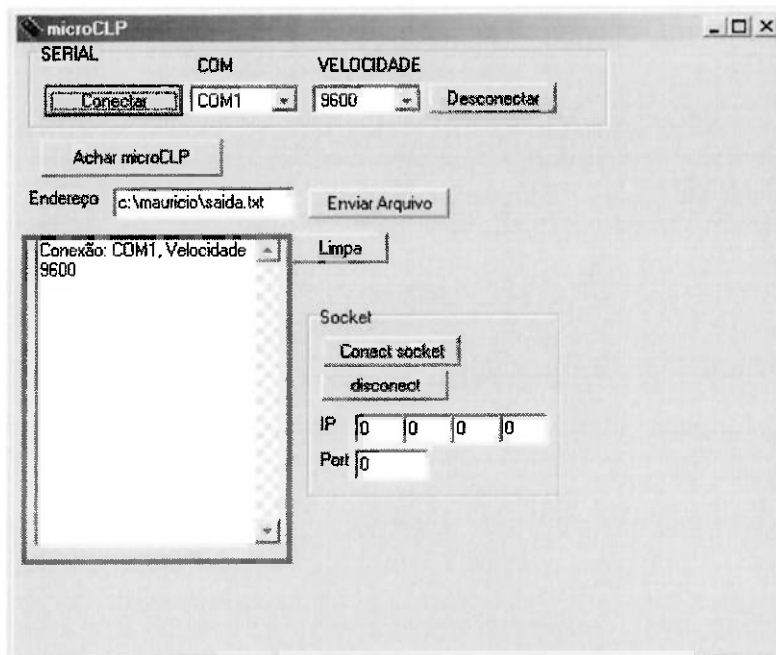


Figura 17. Indicação da abertura da porta COM

2. Clicar no botão “Achar microCLP”

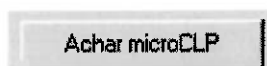


Figura 18. Botão do programa de envio de dados

--	--

O botão serve para verificar se o microCLP (módulo de execução) está conectado à porta COM do computador e se seu status está como “Comunicação”. Obtendo sucesso, o programa gera a seguinte mensagem no campo log: “MICRO-CLP IS ON”

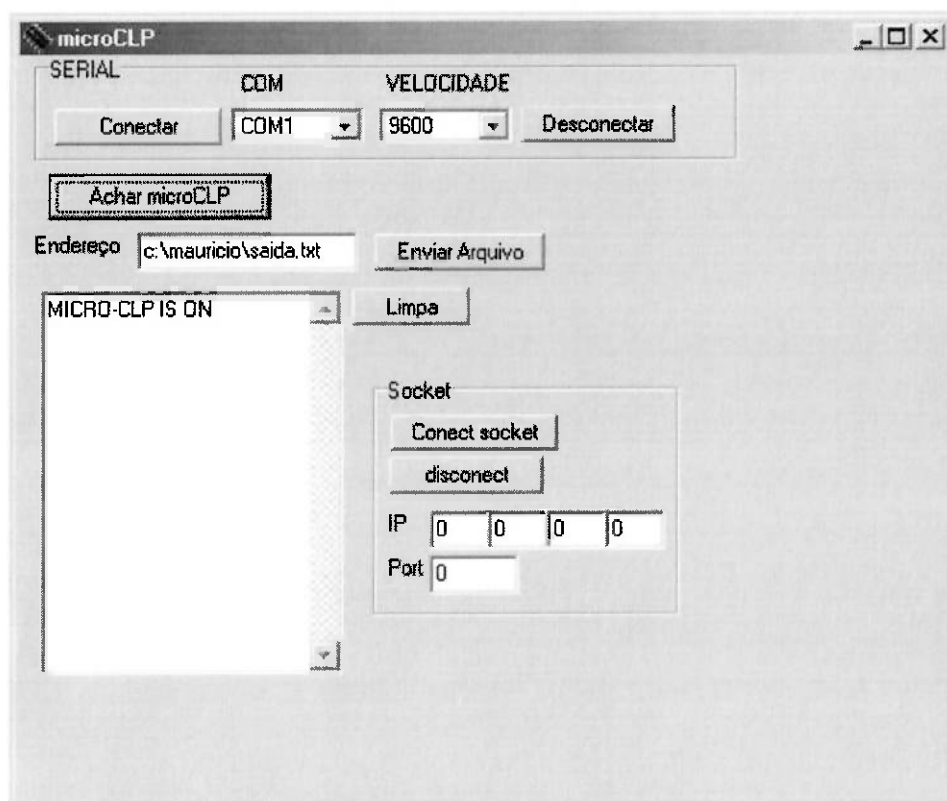


Figura 19. Programa confirmando o sucesso na comunicação

3. Especificar o endereço do diretório aonde o arquivo de saída está localizado. O endereço padrão já está gravado (c:\microCLP\saida.txt).

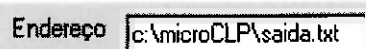


Figura 20. Campo de endereço do arquivo de saída

4. Clicar no botão “Enviar Arquivo”



Figura 21. Mensagem de arquivo não encontrado

A princípio, o programa verifica se o arquivo “Saída.txt” existe no endereço do diretório especificado acima. Caso não, o sistema gera a seguinte mensagem de aviso: “FILE NOT FOUND”

Caso o arquivo esteja no local, o sistema gera a seguinte mensagem no campo log: “Arquivo Saída.txt OK!!!”

Confirmada a existência do arquivo, o programa passa a monitorar o envio dos dados e sua gravação no módulo de execução. Obtendo êxito na gravação, o módulo gera a seguinte mensagem no campo log: “PROGRAMA OK”.

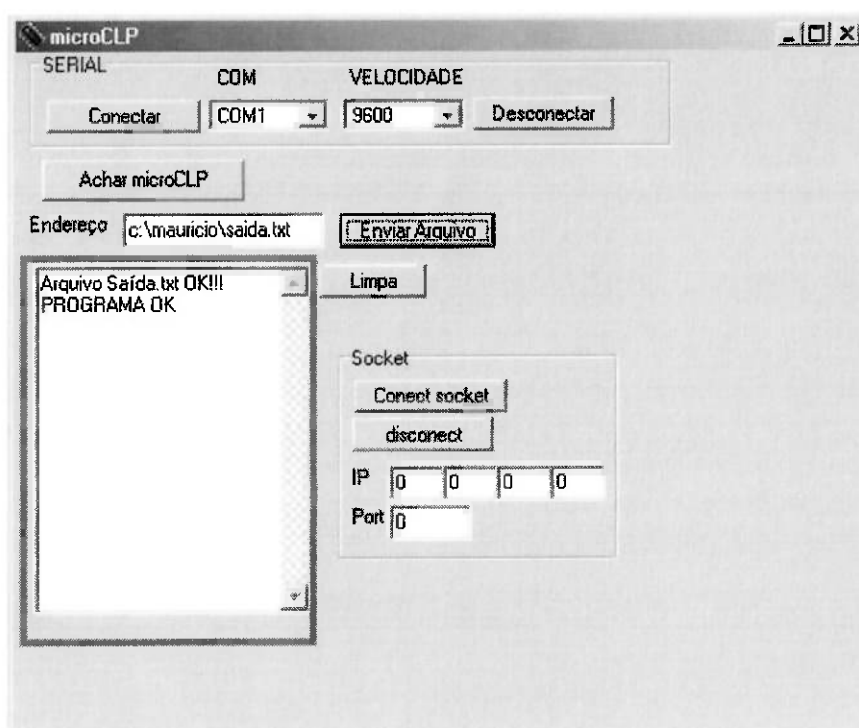


Figura 22. Mensagem de programa carregado no microCLP

Assim, executados os passos acima listados, o programa de controle elaborado na interface está carregado no microCLP e pronto para execução.

INTERFACE DE PROGRAMAÇÃO DO MICROCLP

Manual do usuário

Versão 1.0

TÍTULO DO DOCUMENTO: Manual do usuário - Interface de Controle		AUTOR: Mauricio Kazuo Taniguti
ARQUIVO : Manual do usuario.doc	Ferramenta utilizada : MS Word 97 / 2000	DATE: 11/12/2004

Índice

Introdução	3
Acesso à Interface de Controle.....	4
Menu Principal	5
Estrutura da Interface	6
Descrição dos botões	6
Descrição das linhas	10
Descrição das linhas anexas (Or)	10
Disposição dos comando nas linhas do diagrama.....	10
Descrição do campo de controle da posição momentânea da linha populada	11
Preenchimento automático dos nomes das variáveis.....	11
Regra de validação de cada linha	12
Criando um diagrama de relés.....	13
Criando um contato normalmente aberto	13
Criando um contato normalmente fechado.....	13
Criando um contador	13
Criando um timer	13
Criando uma entrada analógica	13
Pulando uma posição na linha do diagrama	13
Criando um comando em paralelo ao outro(Or).....	14
Limpando uma linha de comando	14
Enviando os dados para o módulo de execução	14

--	--

Introdução

O propósito deste documento é apresentar os procedimentos para a execução de cada tarefa que pode ser realizada na Interface de Programação do microCLP.

O que é a interface?

Uma ferramenta que permite editar programas de controle do microCLP em LD.

--	--

ACESSO À INTERFACE

A interface poderá ser acessada a partir de um atalho criado na área de trabalho do microcomputador do usuário, denominado “Interface_de_Control.xls”. A Figura 1 exibe o atalho da Interface de Programação.



Figura 1. Atalho da Interface de Programação

Menu Principal

A tela do Menu Principal aparece logo após o usuário abrir a interface.

A partir da tela Menu Principal o usuário poderá acessar todas as funções da interface, através da barra de menus ou da barra de ícones de atalho.

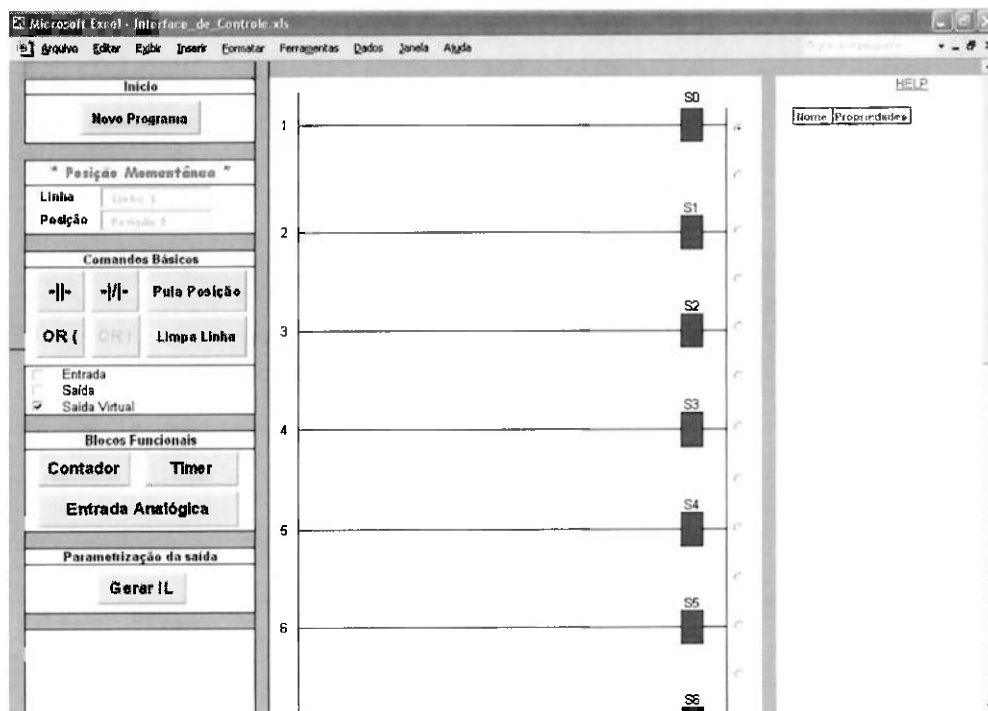


Figura 2. Menu Principal da Interface de Programação

Estrutura da Interface

A interface de controle é estruturada nos seguintes pontos:

- 11 comandos
- 9 linhas de instruções
- 9 linhas anexas (OR)
- 5 posições por linha + 1 posição de saída(bobina)
- Campo de controle da posição momentânea da linha sendo editada
- Nome da variável preenchido automaticamente
- A primeira posição de cada linha válida é obrigatória
- Execução do programa “microCLP”

1- Descrição dos botões de comando:

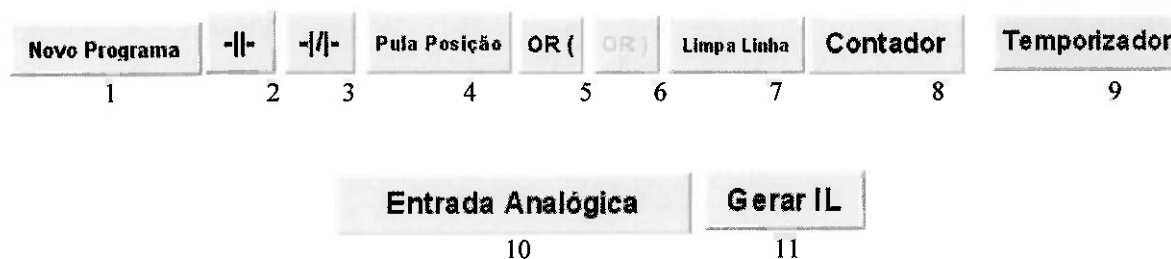


Figura 3. Botões de comando

1. Novo Programa
2. Contato Normalmente Aberto
3. Contato Normalmente Fechado
4. Pula Posição
5. Abre Or
6. Fecha Or
7. Limpa Linha
8. Contador
9. Temporizador
10. Entrada Analógica
11. Gera IL

A. Comando “Novo Programa”

O botão “Novo Programa” permite que o usuário limpe o programa de controle anterior e insira um novo LD.

I – Interface em edição

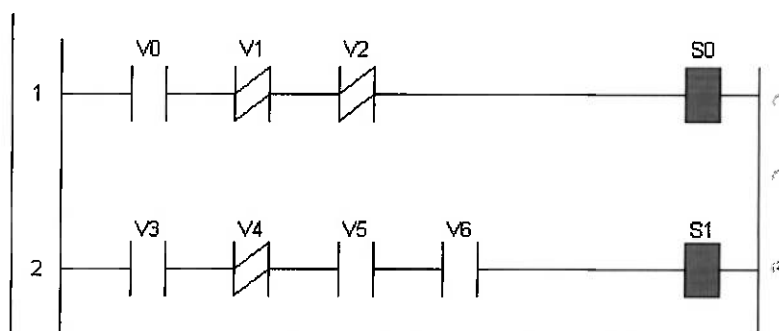


Figura 4. Linhas em edição do LD

2 – Após clicar no botão “Novo Programa” as linhas são limpas

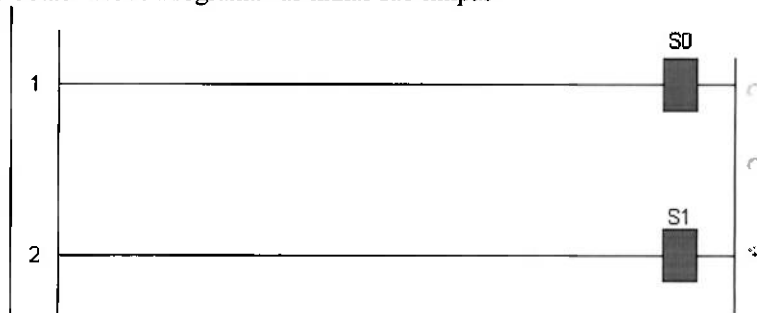


Figura 5. Linhas Limpas do Diagrama em edição

B. Comando “Contato Normalmente Aberto”

O comando “Contato Normalmente Aberto” permite que o usuário insira um contato normalmente aberto no programa de controle.

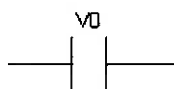


Figura 6. Contato Normalmente Aberto

C. Comando “Contato Normalmente Fechado”

O comando “Contato Normalmente Fechado” permite que o usuário insira um contato normalmente fechado no programa de controle.



Figura 7. Contato Normalmente Fechado

D. Comando “Pula Posição”

O comando “Pula Posição” permite que o usuário pule uma posição na linha em edição.

E. Comando “Abre Or”

O comando “Abre Or” permite que o usuário crie uma nova linha paralela à linha de trabalho atual.

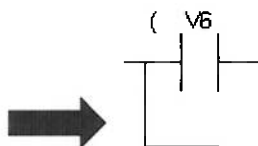


Figura 8. Início da linha OR

F. Comando “Fecha Or”

O comando “Fecha Or” permite que o usuário feche a linha paralela à linha de trabalho atual.

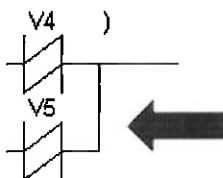


Figura 9. Fim da linha OR

G. Comando “Limpa Linha”

O comando “Limpa Linha” tem a funcionalidade similar ao botão “Novo Programa”. Entretanto, ao invés de permitir que o usuário limpe o programa anterior e insira um novo LD, o botão limpa apenas a linha atual de trabalho.

1 – Interface em edição

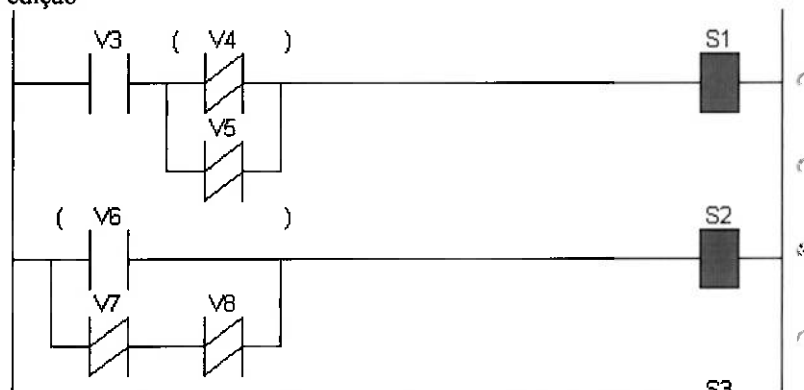


Figura 10. Linha antes de ser apagada

2 – Após clicar no botão “Limpa Linha”

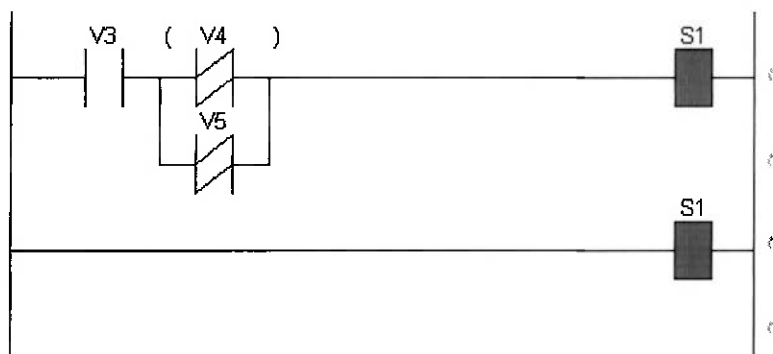


Figura 11. Linha apagada

H. Comando "Contador"

O comando "Contador" permite que o usuário insira um contador no programa de controle.

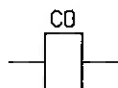


Figura 12. Contato equivalente ao contador

Nome	Propriedades		
CO	0	0	0

Figura 13. Representação das configurações do bloco na interface

Ao inserir um contador no diagrama, no lado direito da tela, aparece uma tabela com o nome da variável e seus parâmetros. Para o contador são 3 parâmetros e inicialmente eles estarão preenchidos com 0 (zero) como valor padrão.

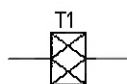
Os parâmetros são:

Tabela 1. Resumo das configurações do bloco contador

Propriedade	Valor	Significado
1	0 -255	Número de incrementos esperado pelo contador
2	48	Incremento na borda de subida do sinal
2	49	Incremento na borda de descida do sinal
3	48	O bloco fecha o contato caso o número esperado seja atingido
3	49	O bloco abre o contato caso o número esperado seja atingido

I. Botão "Temporizador"

O comando "Temporizador" permite que o usuário insira um temporizador no programa de controle.



--	--

Figura 14. Contato equivalente ao temporizador

Nome	Propriedades
T0	0 0 0

Figura 15. Representação das configurações do bloco na interface

Ao inserir um temporizador no diagrama, no lado direito da tela, aparece uma tabela com o nome da variável e seus parâmetros. Para ele são 3 parâmetros e inicialmente eles estarão preenchidos com 0 (zero) como valor padrão.

Os parâmetros são:

Tabela 2. Resumo das configurações do bloco temporizador

Propriedade	Valor	Significado
1	0 -255	Número de incrementos esperada pelo temporizador
2	48	Unidade em segundos
2	49	Unidade em décimos
2	50	Unidade em minutos
3	48	O bloco fecha pelo tempo estipulado e depois abre a saída
3	49	O bloco abre pelo tempo estipulado e depois fecha a saída
3	50	O bloco espero o tempo e depois abre o contato
3	51	O bloco espera o tempo e depois fecha o contato

J. Comando "Entrada Analógica"

O comando "Entrada Analógica" permite que o usuário insira uma entrada analógica no programa de controle.

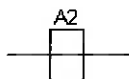


Figura 16. Contato equivalente à entrada analógica

Nome	Propriedades
A0	0 0 0

Figura 17. Representação das configurações do bloco na interface

Ao inserir uma entrada analógica no diagrama, no lado direito da tela, aparece uma tabela com o nome da variável e seus parâmetros. Para a entrada analógica são 2 parâmetros e inicialmente elas estarão preenchidas com 0 (zero) como valor padrão.

Os parâmetros são:

Tabela 3. Resumo das configurações do bloco na interface

Propriedade	Valor	Significado
1	0 -255	Número esperado na entrada analógica
2	48	Sinal cruza o "set point" subindo
2	49	Sinal cruza o "set point" descendo
3	48	O bloco fecha o contato caso o número esperado seja atingido
3	49	O bloco abre o contato caso o número esperado seja atingido

K. Comando “Gera IL”

O botão “Gera IL” permite que o usuário, após editar o programa de controle, enviar este programa para o microCLP.

2 - Descrição das linhas do diagrama:

A linha de instruções é onde o usuário insere operações lógicas.

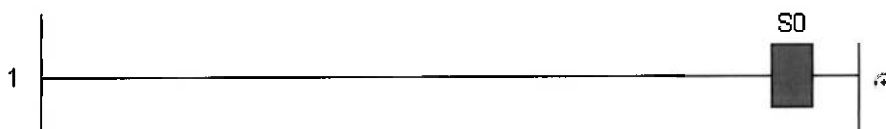



Figura 18. Linha vazia

Para editar a linha, o usuário deve clicar sobre a caixa de opção () presente ao lado da linha.

3 - Descrição das linhas adicionais (Or) do diagrama:

A linha adicional é onde o usuário insere os comandos quando desejar realizar um comando OR.

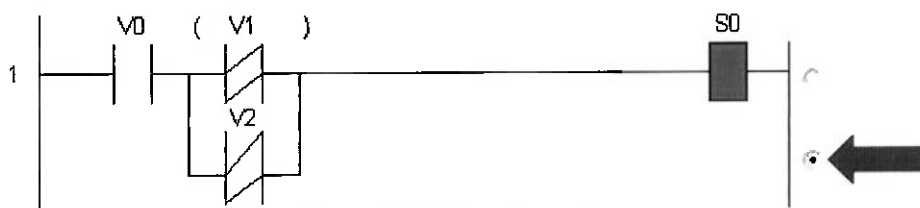



Figura 19. Linha adicional OR

Para editar a linha adicional, o usuário deve clicar sobre a caixa de opção () presente ao lado da linha.

4 – A disposição dos comandos na linha:

Cada uma das linhas (linhas de instrução e linhas anexas) do diagrama possui 5 posições onde as operações são editadas mais uma posição de saída(bobina).

5 – Descrição do campo de controle da posição momentânea da linha em edição:

O campo de controle da posição momentânea da linha em edição permite que o usuário saiba qual posição da linha está sendo editada no momento.

* Posição Momentânea *	
Linha	Or - Linha 1
Posição	Posição 3

Figura 20. Indicação da posição e linha

6 – Preenchimento automático dos nomes das variáveis:

Ao editar uma linha do diagrama com um contato, um temporizador, um contador ou uma entrada analógica, o nome de sua variável é preenchido automaticamente, de acordo com a seguinte regra:

- ☐ Entrada
- ☐ Saída
- ☒ Saída Virtual

Figura 21. Caixas de seleção para opção de nome das variáveis

1. Caso a opção “Entrada” esteja selecionada :

Para que o nome da variável seja uma entrada, basta o usuário clicar na caixa de seleção acima, presente no menu principal, a opção “Entrada”. Assim, o nome de cada variável é preenchido com a letra E.

2. Caso a opção “Saída” esteja selecionada:

Para que o nome da variável seja uma saída, basta o usuário clicar na caixa de seleção acima, presente no menu principal, a opção “Saída”. Assim, o nome de cada variável é preenchido com a letra S.

3. Caso a opção “Saída Virtual” esteja selecionada:

Para que o nome da variável seja uma saída virtual, basta o usuário clicar na caixa de seleção acima, presente no menu principal, a opção “Saída Virtual”. Assim, o nome de cada variável é preenchido com a letra V.

Caso o usuário queira alterar o nome da variável de algum contato já inserido, basta alterar o nome da que aparece sobre o contato desenhado no diagrama.

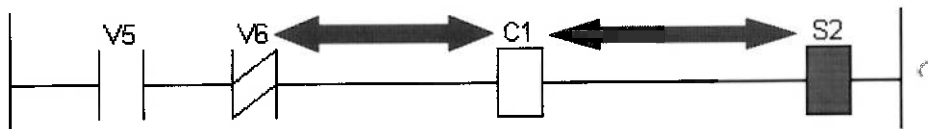


Figura 22. Saídas virtuais

7 – Regra de validação de cada linha:

Para que uma linha seja válida e compilada, sua primeira posição deve estar preenchida com um contato válido.

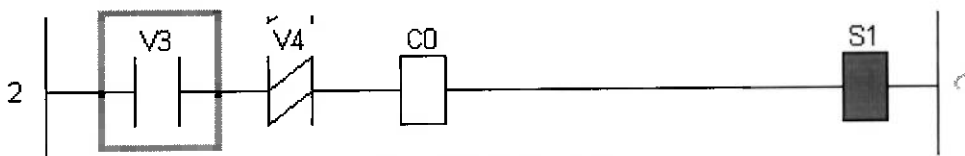


Figura 23. Linha válida

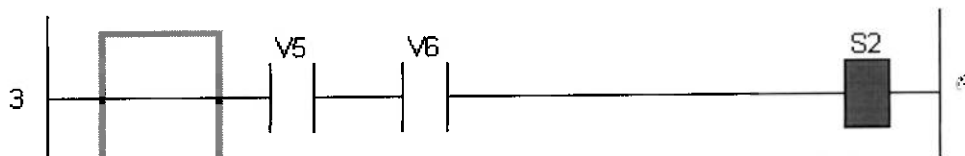


Figura 24. Linha inválida (1ª posição não está com contato válido)

8 – Execução do programa “microCLP”:

Clicando no botão “Gerar IL”, a interface abre o programa “microCLP” (microCLP.exe) que monitora, loga, e envia os dados do arquivo texto para o módulo de execução. Para o envio dos dados para o módulo de execução, deve-se executar os seguintes passos:

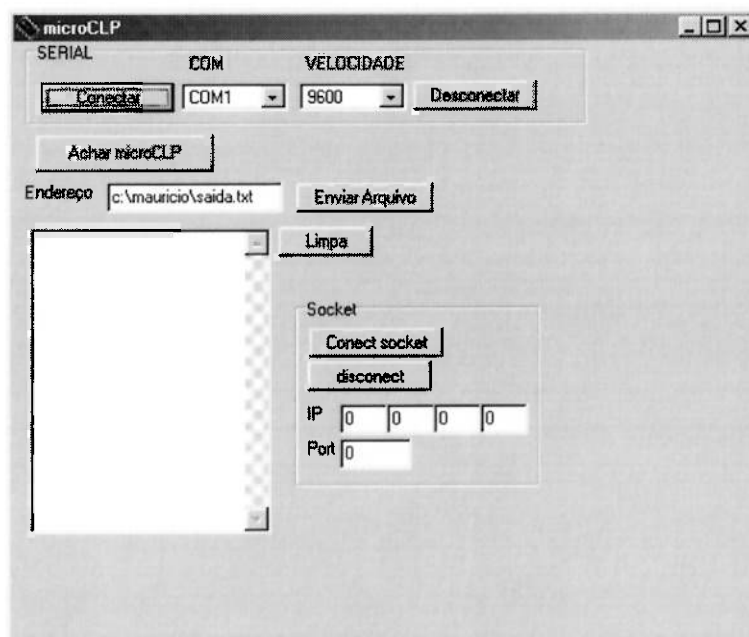


Figura 25. Exemplo do programa de envio de dados

1. Clicar no botão “Conectar”



Figura 26. Botão do programa de envio de dados

No campo de log, o programa mostra o status da conexão com a porta COM do computador.

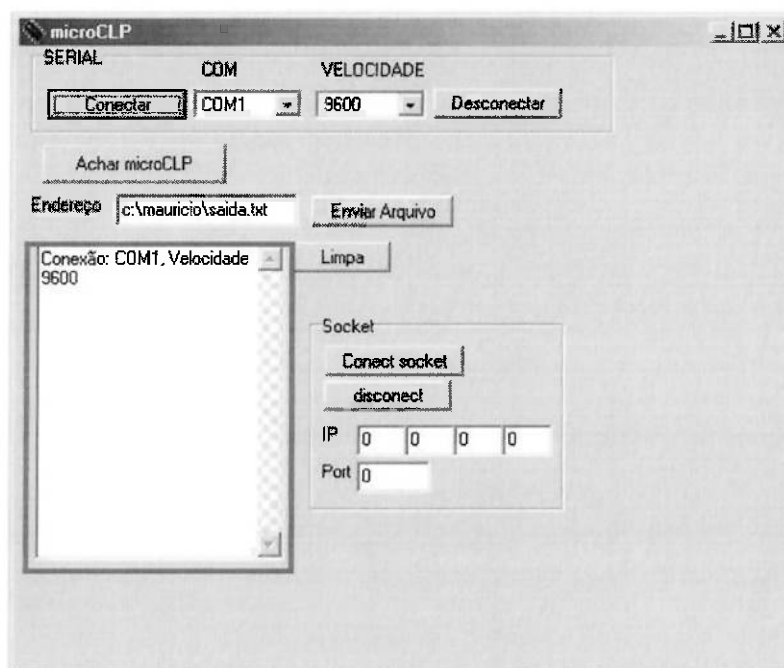


Figura 27. Indicação da abertura da porta COM

2. Clicar no botão “Achar microCLP”

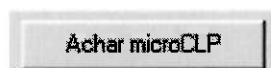


Figura 28. Botão do programa de envio de dados

O botão serve para verificar se o microCLP está conectado à porta COM do computador e se seu status está como “Comunicação”. Obtendo sucesso, o programa gera a seguinte mensagem no campo log: “MICRO-CLP IS ON”

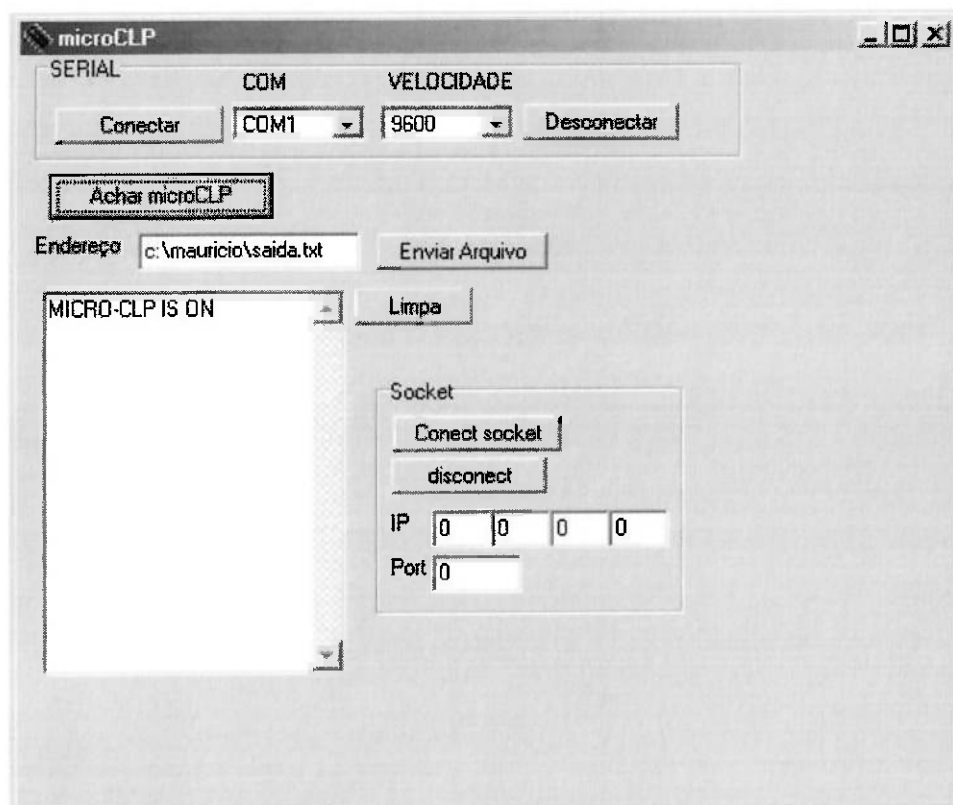


Figura 29. Programa confirmando o sucesso na comunicação

3. Especificar o endereço do diretório aonde o arquivo de saída está localizado. O endereço padrão já está gravado (c:\microCLP\saida.txt).



Figura 30. Campo de endereço do arquivo de saída

4. Clicar no botão “Enviar Arquivo”



Figura 31. Mensagem de arquivo não encontrado

A princípio, o programa verifica se o arquivo “Saida.txt” existe no endereço do diretório especificado acima. Caso não, o sistema gera a seguinte mensagem de aviso: “FILE NOT FOUND”

Caso o arquivo esteja no local, o sistema gera a seguinte mensagem no campo log: “Arquivo Saída.txt OK!!!”

Confirmada a existência do arquivo, o programa passa a monitorar o envio dos dados e sua gravação no módulo de execução. Obtendo êxito na gravação, o módulo gera a seguinte mensagem no campo log: “PROGRAMA OK”.

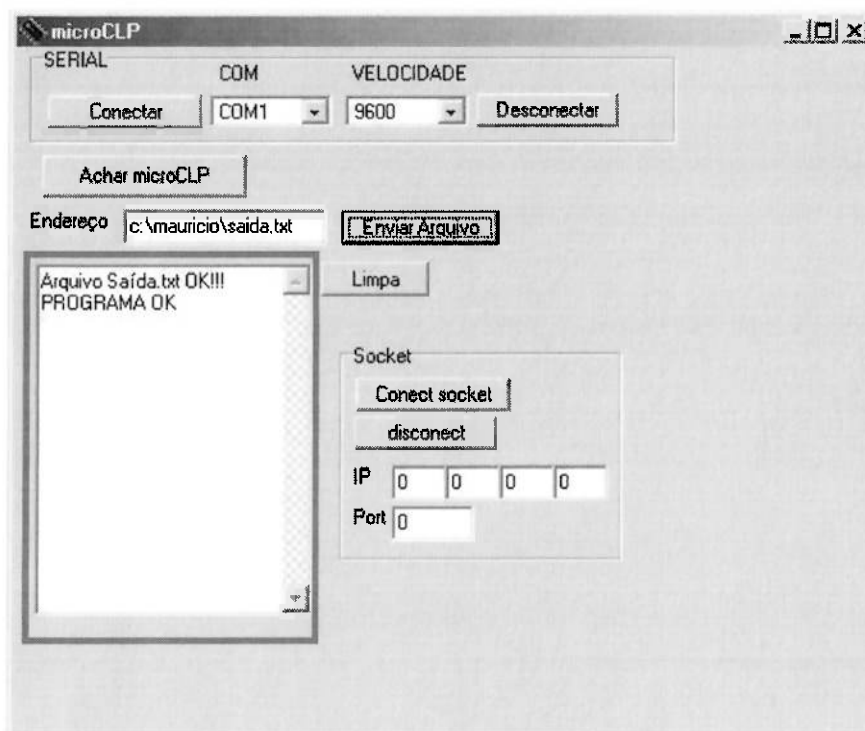



Figura 32. Mensagem de programa carregado no microCLP

Assim, executados os passos acima listados, o programa de controle em LD criado na interface de programação está carregado no microCLP e pronto para execução.

CRIANDO UM DIAGRAMA DE RELÉS

Para a criação de um diagrama de relés, o usuário deve seguir os seguintes passos:

- 1 – Ao abrir o arquivo, clicar sobre o botão “Novo Programa”
- 2 – Clicar sobre a caixa de opção () da linha que se deseja popular


Criando um contato normalmente aberto

- 3 – Basta clicar sobre o botão contato normalmente aberto ()


Criando um contato normalmente fechado

- 4 – Basta clicar sobre o botão contato normalmente fechado ()


Criando um contador

- 5 – Basta clicar sobre o botão contador ()


Criando um temporizador

- 6 – Basta clicar sobre o botão temporizador ()

Criando uma entrada analógica


- 7 – Basta clicar sobre o botão entrada analógica ()


Pulando uma posição na linha do diagrama

- 8 – Basta clicar sobre o botão pula posição ()


Criando um comando em paralelo a outro (Or)

--	--


8 – Clicar no botão “Abre Or” ()

9 – Clicar sobre a caixa de opção () presente ao lado da linha adicional que é cosntruída.

10 – Editar a linha adicional com operadores a serem considerados (obrigatório).


11 – Clicar no botão “Fecha OR” ()


Limpando uma linha de comando

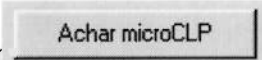
12 – Caso o usuário insira um comando errado, para apagar apenas a linha, ele deve clicar na caixa de opção () da linha desejada.


13 – Clicar no botão “Limpa Linha” ()

Enviando os dados para o módulo de execução

14 – Após popular as linhas desejadas, basta clicar no botão “Gerar IL” () e o programa microCLP é aberto.

15 – No programa, clicar sobre o botão “Conectar” ()

16 – No programa, clicar sobre o botão “Achar microCLP” ()

17 - Confirmada a conexão com o microCLP, especificar o endereço do diretório onde o arquivo de saída está localizado ( c:\microCLP\saida.txt). O endereço padrão já está gravado (c:\microCLP\saida.txt).

18 –Especificada a saída, clicar sobre o botão “Enviar Arquivo” ().

Início

Novo Programa

* Posição Momentânea *

Linha

Posição

Comandos Básicos

Pula Posição

OR ()

Limpa Linha

☐ Entrada
 ☐ Saída
 ☒ Saída Virtual

Blocos Funcionais

Contador

Timer

Entrada Analógica

Parametrização da saída

Gerar IL

HELP

Nome	Propriedades			
C0	0	0	0	0
T0	0	0	0	0
A1	0	0		
C1	0	0	0	0

1

V0

(V1)

V2

S0

2

V3

V4

C0

T0

A1

S1

3

V5

V6

C1

S2

4

S3

5

S4

6

S5

Linhas do diagrama de relés em edição

Gerar IL

Após clicar no botão, o programa de controle em LD é compilado, transformado para lista de instrução, decimal, ASCII e os dados são enviados para o módulo de execução.