

**UNIVERSIDADE SÃO FRANCISCO**  
**CURSO DE ENGENHARIA ELÉTRICA**

**CONTROLE E INTEGRAÇÃO  
DE UM POSTO DE TRABALHO**

Área de Automação

por

Rodrigo Pinheiro Martos

Cíntia Kimie Aihara, Mestre  
Orientador

Itatiba (SP), novembro de 2004

**UNIVERSIDADE SÃO FRANCISCO**  
**CURSO DE ENGENHARIA ELÉTRICA**

**CONTROLE E INTEGRAÇÃO  
DE UM POSTO DE TRABALHO**

Área de Automação

por

Rodrigo Pinheiro Martos

Relatório apresentado à Banca Examinadora do  
Trabalho de Conclusão do Curso de Engenharia  
Elétrica para análise e aprovação.  
Orientador: Cíntia Kimie Aihara, Mestre.

Itatiba (SP), novembro de 2004

# SUMÁRIO

<b>LISTA DE ABREVIATURAS .....</b>	<b>iv</b>
<b>LISTA DE FIGURAS .....</b>	<b>v</b>
<b>LISTA DE TABELAS.....</b>	<b>vi</b>
<b>RESUMO.....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
<b>1.1. OBJETIVOS.....</b>	<b>2</b>
<b>1.1.1. Objetivo Geral.....</b>	<b>2</b>
<b>1.1.2. Objetivos Específicos.....</b>	<b>2</b>
<b>1.2. METODOLOGIA.....</b>	<b>3</b>
<b>1.3. ESTRUTURA DO TRABALHO .....</b>	<b>3</b>
<b>2. FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>4</b>
<b>2.1. CONTROLADOR LÓGICO PROGRAMAVEL (CLP).....</b>	<b>4</b>
<b>2.2. LINGUAGEM GRAFCET.....</b>	<b>5</b>
<b>3. PROJETO .....</b>	<b>8</b>
<b>3.1. SOFTWARE WINSPS.....</b>	<b>8</b>
<b>3.1.1. Contato normal aberto (N/A) .....</b>	<b>9</b>
<b>3.1.2. Contato normal fechado (N/F).....</b>	<b>9</b>
<b>3.1.3. Saída (X).....</b>	<b>9</b>
<b>3.1.4. Set (S).....</b>	<b>10</b>
<b>3.1.5. Reset (R).....</b>	<b>10</b>
<b>3.1.6. Timer - Pulse (SP).....</b>	<b>11</b>
<b>3.1.7. Timer - Extended Pulse (SV).....</b>	<b>11</b>
<b>3.1.8. Timer - Stop (TH).....</b>	<b>12</b>
<b>3.1.9. Timer - Reset (RT).....</b>	<b>12</b>
<b>3.1.10. Contador Set.....</b>	<b>13</b>
<b>3.2. PROGRAMANDO O CLP .....</b>	<b>13</b>
<b>3.2.1. 1º Passo.....</b>	<b>15</b>
<b>3.2.2. 2º passo:.....</b>	<b>15</b>
<b>3.2.3. 3º passo.....</b>	<b>16</b>
<b>3.2.4. 4º passo.....</b>	<b>16</b>
<b>3.2.5. 5º passo.....</b>	<b>16</b>
<b>3.3. VALIDAÇÃO .....</b>	<b>17</b>
<b>3.3.1. Etapas 0 e 1.....</b>	<b>19</b>
<b>3.3.2 Etapas 2 e 3.....</b>	<b>20</b>
<b>3.3.3. Etapas 4, 5 e 6.....</b>	<b>22</b>
<b>3.3.4. Etapas 7, 8, 9 e 10.....</b>	<b>24</b>

<b>3.3.5. Etapa 11.....</b>	<b>26</b>
<b>3.4. SIMULAÇÃO .....</b>	<b>27</b>
<b>CONSIDERAÇÕES FINAIS.....</b>	<b>31</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>32</b>

## **LISTA DE ABREVIATURAS**

CLP                      Controlador Lógico Programável

## LISTA DE FIGURAS

Figura 1. Controlador Lógico Programável (CLP). .....	4
Figura 2. Interface do Software. ....	8
Figura 3. Possíveis conexões. ....	9
Figura 4. Simbologia do contato N/A. ....	9
Figura 5. Simbologia do contato N/F. ....	9
Figura 6. Simbologia da bobina. ....	10
Figura 7. Simbologia da bobina “setada”. ....	10
Figura 8. Simbologia da bobina de reset. ....	11
Figura 9. Simbologia do Timer. ....	11
Figura 10. Demonstrativa de como o loop do Timer funciona. ....	11
Figura 11. Simbologia do Timer. ....	12
Figura 12. Demonstrativa de como o loop do Timer funciona. ....	12
Figura 13. Simbologia do Timer de stop. ....	12
Figura 14. Simbologia do Timer de reset. ....	12
Figura 15. Simbologia do contador. ....	13
Figura 16. Antes da otimização. ....	13
Figura 17. Depois da otimização. ....	13
Figura 18. CL 150. ....	14
Figura 19. CLP alimentado pronto para ser usado. ....	15
Figura 20. CLP conectado na porta serial. ....	15
Figura 21. Confirmação de comunicação. ....	16
Figura 22. Chave STOP\RUN. ....	16
Figura 23. Acompanhamento simultâneo. ....	17
Figura 24. Layout do posto de trabalho. ....	17
Figura 25. Modelo do posto de trabalho. ....	18
Figura 26. Etapas 0 e 1. ....	19
Figura 27. Elementos utilizados nas etapas 0 e 1. ....	19
Figura 28. Programa relacionado à etapa 0 e 1. ....	20
Figura 29. Etapas 2 e 3. ....	20
Figura 30. Elementos utilizados nas etapas 2 e 3. ....	21
Figura 31. Programa relacionado à etapa 2 e 3. ....	22
Figura 32. Etapas 4, 5 e 6. ....	22
Figura 33. Elementos utilizados nas etapas 4, 5 e 6. ....	23
Figura 34. Programa relacionado à etapa 4, 5 e 6. ....	23
Figura 35. Etapas 7, 8, 9 e 10. ....	24
Figura 36. Elementos utilizados nas etapas 7, 8, 9 e 10. ....	24
Figura 37. Programa relacionado à etapa 7, 8, 9 e 10. ....	25
Figura 38. Programa do Contador. ....	26
Figura 39. Etapa 11. ....	26
Figura 40. Programa que encerra o processo. ....	27
Figura 41. Fonte ajustada para alimentar o CLP. ....	28
Figura 42. Entrada I1.0 sendo acionado. ....	29
Figura 43. Saída Q0.0 ativada. ....	29
Figura 44. Motor utilizado na simulação. ....	30
Figura 45. Relê utilizado para controle do motor. ....	30

## **LISTA DE TABELAS**

Tabela 1. Descrição das abreviaturas utilizadas no modelo.....	18
--	----

## **RESUMO**

MARTOS, Rodrigo Pinheiro. Controle e Integração de um Posto de Trabalho. Itatiba, 2004. Trabalho de Conclusão de Curso, Universidade São Francisco, Itatiba, 2004.

Este trabalho trata-se da integração e controle dos diversos elementos que compõem de um posto de trabalho através da utilização de um controlador lógico programável (CLP). Através da simulação este trabalho visa demonstrar a automação de um posto de trabalho para corte de cabo. Esse posto de trabalho é composto por atuadores, chaves fim de curso e um motor de corrente contínua. Relés são utilizados para controlar o sentido de rotação do motor. Utilizamos também temporizadores e contadores para o controle do posto de trabalho. O software utilizado para desenvolver o programa do CLP é o WinSPS. Esse fornece muitos recursos como contadores, temporizadores, e outros. A linguagem de programação predominante é a Ladder que nos permite uma visualização mais fácil do processo.

**Palavras-chave:** Automação Industrial, Integração de Sistemas, Controlador Lógico Programável.

## **ABSTRACT**

This project is about the controlling and integration of a work place having a Programmable Logic Controller (PLC) as a factor responsible for integrating the electronic components. This work place will be composed of actuators, end micro switch, and a direct current engine. Relés will be used to control the rotating direction of the engine due to the missing analogical output in the PLC. Counters and timers will also be used to the work place control. The software used to generate the PLC program will be WINMSPS. This software provides many resources such as counters, timers and others. The predominant programming language will be Ladder, which allows one to have a better visualization of the process.

***Keywords:*** *Industry automation, systems integration, Programmable Logic Controller.*

# 1. INTRODUÇÃO

A automação industrial se verifica sempre que novas técnicas de controle são induzidas num processo. Associado ao aumento de produtividade, pode-se dizer que automação industrial se propõe a oferecer e gerenciar soluções, pois ela sai do nível chão de fábrica para voltar seu foco para o gerenciamento de informações. Devemos notar a diferença, mesmo que sutil, entre automatização e automação [1].

A automatização está ligada à sugestão de movimento automático, repetitivo, mecânico, portanto reproduz uma ação. Nesse caso tem –se um sistema que a saída independe da entrada, ou seja, não existe uma relação entre o valor desejado e o valor recebido. Diz-se que esse tipo de controle se dá por malha aberta [1].

Já a automação é um conceito e conjuntos de técnicas por meio das quais se constroem sistemas ativos capazes de atuar com uma eficiência ótima pelo uso de informações recebidas do meio sobre o qual atuam. Com base nas informações, o sistema calcula a ação corretiva mais apropriada para a execução da próxima ação. Esta é a característica de um sistema de malha fechada, conhecidos como sistemas de realimentação, ou seja, mantém uma relação entre a entrada e a saída. Essa relação entrada/saída serve para corrigir eventuais valores na saída que estejam fora dos valores desejados. Para tanto, são utilizados controladores que por meio da execução algorítmica de um programa compara o valor atual com o desejado. O valor desejado também é conhecido como setpoint [1].

O CLP (controlador lógico programável) é capaz de armazenar instruções para implementação de funções de controle (seqüência lógica, temporização e contagem), além de realizar operações lógicas e aritméticas, manipulação de dados e comunicação em rede. Ele é composto por quatro blocos principais: CPU, Circuito/Modulo de I/O, fonte de alimentação e a base [1].

A CPU é responsável por executar a leitura dos status dos dispositivos de entrada por meio dos módulos de I/O. Esses status são armazenados na memória RAM para serem processados pelo Programa de Aplicação (desenvolvido pelo usuário e armazenado na memória do CLP). Após a execução do programa de aplicação, o processador atualiza os status dos dispositivos de saída, realizando a lógica de controle [1].

Atualmente, os CLPs apresentam características como módulos de I/O de alta densidade, módulos remotos controlados por uma mesma CPU, módulos inteligentes (controle PDI, posicionamento de eixos, transmissão via rádio ou modem, leitura de códigos de barras), recursos de monitoramento da execução do programa podendo detectar eventuais falhas, tamanhos menores que oferecem recursos de hardware e software cada vez maiores e softwares de programação em ambiente Windows [1].

A programação do CLP é feita por meio de uma ferramenta de programação, que pode ser um programador manual ou um computador com software de programação específico. A linguagem Ladder é a mais utilizada por ser muito parecida com a lógica de contatos de relés [1].

Mesmo tendo sido a primeira linguagem destinada à programação de CLPs, essa ainda é a mais utilizada e se encontra em praticamente todos os CLPs disponíveis no mercado. Por ser uma linguagem gráfica baseada em símbolos semelhantes aos encontrados nos esquemas elétricos (contatos e bobinas) as possíveis diferenças existentes entre fabricantes são facilmente assimiladas pelos usuários [1].

Cada programa de aplicação é composto por lógicas de controle, que por sua vez é dividida em rungs compostos de linhas e colunas. A quantidade de linhas e colunas é determinada pelos fabricantes, podendo variar conforme a CPU utilizada. Cada elemento da lógica de controle representa uma instrução da linguagem Ladder, sendo alocada em um endereço específico e armazenada no Programa de Aplicação [1].

Dentro deste contexto este trabalho é desenvolvido, utilizando idéias de automatização e a utilização de um CLP para facilitar a integração dos componentes eletrônicos.

## **1.1. OBJETIVOS**

### **1.1.1. Objetivo Geral**

O objetivo desse projeto é obter o controle de um posto de trabalho e realizar a integração entre vários elementos constituídos de atuadores, chaves fim de curso e um motor elétrico.

### **1.1.2. Objetivos Específicos**

Para alcançar o objetivo geral de controlar um posto de trabalho foram estabelecidos alguns objetivos específicos citados abaixo.

- ?? Levantamento e conhecimento dos componentes que serão utilizados no desenvolvimento do projeto;
- ?? Conhecimento da linguagem Ladder, visto que o software de programação do CLP disponibiliza a mesma como ferramenta de trabalho;
- ?? Desenvolver habilidades para a utilização do CLP;
- ?? Desenvolver o sistema de transferência e transporte de material.

## **1.2. METODOLOGIA**

Levantamento dos componentes a serem utilizados, estudo e pesquisas em livros e sites, criação de programas de aplicação em linguagem Ladder, simulação com a utilização do CLP e implementação.

## **1.3. ESTRUTURA DO TRABALHO**

O trabalho é estruturado da seguinte forma:

1. Introdução: neste capítulo é realizado um breve histórico da automação e as delimitações do trabalho.
2. Fundamentação teórica: nesta etapa do trabalho são discutidos conceitos sobre automação e controlador lógico programável e conhecimentos sobre a linguagem Grafcet.
3. Projeto: são apresentados informações importantes sobre o software WinSPS (recursos, características e facilidades), roteiro passo a passo para programar o controlador lógico programável e a validação do trabalho, apresentando seu modelo e programação.
4. Considerações finais: conclusões e resultados obtidos no desenvolvimento do trabalho.

## 2. FUNDAMENTAÇÃO TEÓRICA

A seguir será apresentado o conhecimento teórico necessário para o desenvolvimento do trabalho. Um estudo sobre controladores lógicos programáveis será mostrado juntamente com um breve histórico do mesmo. Conceitos da linguagem Grafcet também serão apresentados para um melhor entendimento mais a frente.

### 2.1. CONTROLADOR LÓGICO PROGRAMAVEL (CLP)

O Controlador Lógico Programável (CLP) foi Idealizado nos Estados Unidos da América, ao final da década de 1960, pela industria automobilística, que na época tinha a necessidade de criar um elemento de controle versátil e, ao mesmo tempo, com uma rápida capacidade de modificação de sua programação.

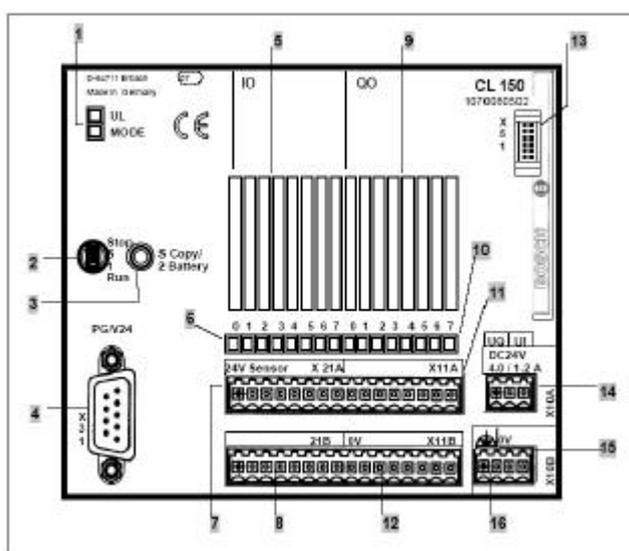


Figura 1. Controlador Lógico Programável (CLP).

Fonte: Manual técnico do CL-150

O CLP pode receber ou enviar informações para o processo através de sinais classificados como sinais digitais ou sinais analógicos.

Os sinais de entradas digitais são aqueles que possuem a função de indicar qualquer ocorrência no processo através de sinais definidos como ligados ou desligados. Por exemplo, um

relê de sobrecarga atuado; sensores ou chaves fim-de-curso; botões para indicar operações a serem efetuadas; entre outros. As entradas podem receber tensões de diversos valores e características conforme a especificação de cada fabricante. O padrão mais comum é encontrar-se sinais de 24 Volts em corrente contínua ou 110 Volts em corrente alternada.

Como exemplo de saídas digitais podem ser citados todos os elementos que são atuados pelo controlador programável, através de um contato ou elemento de estado sólido, como um tiristor que através do programa pode-se acionar um elemento do processo. Como estas saídas possuem limitação de corrente de operação, utiliza-se ligados a elas, elementos como, por exemplo, uma chave magnética responsável pelo acionamento de um motor elétrico ou qualquer outra carga que consuma grande energia; relês para o acionamento de cargas menores como bobinas eletromagnéticas de acionamento de válvulas pneumáticas ou hidráulicas; ou cargas de baixo consumo, como lâmpadas e sirenes ligadas diretamente a saída.

Muitas vezes em um processo, não basta apenas saber se um elemento foi acionado ou não, mas o quanto foi acionado. Para estas situações utilizam-se sinais de entrada analógicos, que são sinais que indicam um valor de uma variável através de um sinal de tensão (0 a 10 Vcc; -5 V a +5 V) ou de corrente proporcional à grandeza medida.

Os sinais analógicos de entradas, ao serem recebidos pelo CLP, são convertidos em números binários, e se diferenciam também com respeito a sua precisão; isto pode ser indicado pelo número de bits composto pelo valor obtido.

Portanto, devido à capacidade de trabalhar com qualquer tipo de sinal, pode-se dizer que um CLP é o elemento ideal para se controlar um sistema ou processo, seja ele analógico ou digital.

## **2.2. LINGUAGEM GRAFCET**

Um Grafcet é um modelo de representação gráfica do comportamento da parte de comando de um sistema automatizado. Ele é constituído por uma simbologia gráfica com arcos orientados que interligam etapas e transições por uma interpretação das variáveis de entrada e saída da parte de comando caracterizadas como receptividades e ações, como mostra [1].

### **2.2.1. Etapas**

Uma etapa significa um estado no qual o comportamento do circuito de comando não se altera frente a suas entradas e saídas. Elas devem ser identificadas com números, seguidos ou não por abreviatura.

Em um determinado instante, uma etapa pode estar ativa ou inativa. O conjunto de etapas ativas em um determinado instante mostra a situação em que o sistema se encontra. A etapa que se torna ativa logo após o início de funcionamento do sistema, é chamada de etapa inicial [1].

### **2.2.2. Transição**

É representada por um traço perpendicular aos arcos orientados e significa a possibilidade de evolução do Grafset de uma situação para outra. Uma transição pode, em um dado instante, encontrar-se válida ou não, sendo que ela é dita válida quando todas as etapas imediatamente precedentes e ligadas a ela estiverem ativas. A passagem de uma situação à seguinte, portanto só é possível com a validade de uma transição, momento esse em que se diz que ocorre a transição [1].

### **2.2.3. Arcos orientados**

Indicam a sequencialização do Grafset pela interligação de uma etapa a uma transição e desta a outra etapa sucessivamente. A interpretação é de cima para baixo [1].

### **2.2.4. Ação**

As ações representam os efeitos que devem ser obtidos sobre o mecanismo controlado em uma determinada situação, ou seja, “o que se deve ser feito”. Cada ação é representada graficamente no interior de retângulos associados a uma etapa, e esta será realizada somente e apenas quando sua etapa correspondente estiver ativa.

De uma forma geral, as ordens de comando contidas em ações podem atuar sobre elementos físicos do mecanismo controlado (saídas do CLP), sobre elementos auxiliares do comando (temporizadores, contadores, memórias, etc), ou ainda em interface homem-maquina (vídeos, painéis de controle, impressora, etc) [1].

### **2.2.5 Receptividade**

Receptividade é a função lógica combinacional associada a cada transição. Quando em estado lógico verdadeiro, uma receptividade irá ocasionar a ocorrência de uma transição válida. Ela pode então ser encarada como um elo de ligação existente entre a lógica combinacional e a sequência lógica.

Na prática uma receptividade pode representar variáveis lógicas tais como sinais de entrada do sistema, variáveis internas de controle, resultado de comparações com contadores e temporizadores, informações sobre o estado de uma etapa, ou ainda condicionada a uma determinada situação do Grafset [1].

### 3. PROJETO

Para o desenvolvimento do trabalho proposto será utilizada, em sua maioria a linguagem de programação Ladder. A seguir teremos uma breve apresentação do software WINSPS que será utilizado para programar o CLP.

#### 3.1. SOFTWARE WINSPS

A figura 2 mostra como os elementos de entrada e saída ficam distribuídos na programação em Ladder.

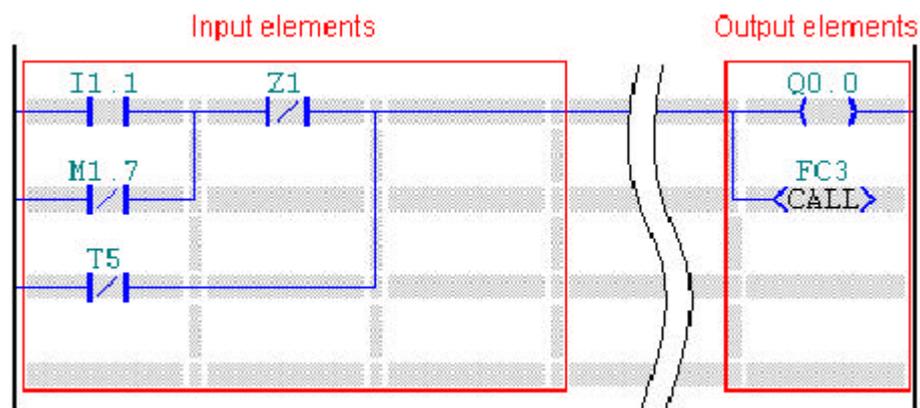


Figura 2. Interface do Software.

O software apresenta desde a utilização de contatos N/A e N/F até elementos de saída mais complexos que serão apresentados abaixo:

Esta função introduz uma linha conectando. As conexões podem estar em horizontal e o sentido vertical, mas o fluxo é sempre da esquerda para a direita. Uma conexão horizontal passa o estado binário de um elemento ao subsequente elemento (da esquerda para a direita!). Isto corresponde a uma conexão AND. Uma conexão vertical conecta todos os elementos horizontais no lado esquerdo com uma combinação OU e passa este valor a todos os elementos do lado direito.

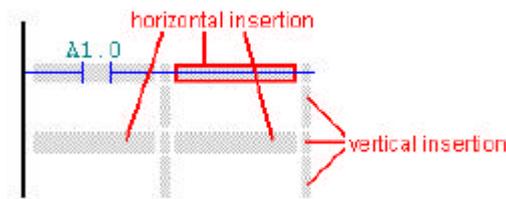


Figura 3. Possíveis conexões.

### 3.1.1. Contato normal aberto (N/A)

Sua função é inserir, como já diz o nome, um contato aberto. Esse contato passa o valor binário existente para a saída [4]. Simbologia:



Figura 4. Simbologia do contato N/A.

### 3.1.2. Contato normal fechado (N/F)

Sua função é inserir, como já diz o nome, um contato fechado. Esse contato passa o valor binário existente em forma negativa para a saída [4]. Simbologia:



Figura 5. Simbologia do contato N/F.

### 3.1.3. Saída (X)

Esta função introduz uma atribuição. A atribuição (da saída) atribui valor da esquerda à variável. Uma bobina fica situada sempre no lado direito do diagrama da escada como o último

elemento diretamente antes do buscar direito. Diversos elementos de saída podem ser introduzidos um abaixo do outro e interconectados [4].



Figura 6. Simbologia da bobina.

### 3.1.4. Set (S)

Esta função introduz uma bobina "setada". Se o estado ON (TRUE) estiver atual na conexão à esquerda da bobina "setada" (S), a variável será ajustada também a ON. Este estado é mantido até que seja cancelado por uma bobina da reset (R) (OFF). Uma bobina é ficada situada sempre no lado direito do diagrama da escada como último elemento. Diversos elementos de saída podem ser introduzidos um abaixo do outro e ser interconectado [4].



Figura 7. Simbologia da bobina "setada".

### 3.1.5. Reset (R)

Esta função introduz uma bobina da reset. Se o estado ON (TRUE) estiver atual sobre a conexão à esquerda de uma bobina da reset (R), a variável será ajustado OFF. Isto faz possível cancelar um estado criado pela bobina "setada" (S). Uma bobina é ficada situada sempre no lado direito do diagrama da escada como último elemento. Diversos output elementos pode ser introduzido um abaixo do outro e ser interconectado [4].

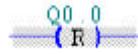


Figura 8. Simbologia da bobina de reset.

### 3.1.6. Timer - Pulse (SP)

O Timer é iniciado com a transição do pulso do nível lógico 1 para 0 como mostra a (10). Após a definição do nome do Timer o usuário define a duração ou tempo do loop, em segundos, como podemos ver na (9). O Timer só é acionado quando a condição de start fica em nível lógico alto por algum tempo, para evitar seu acionamento por ruídos [4].



Figura 9. Simbologia do Timer.

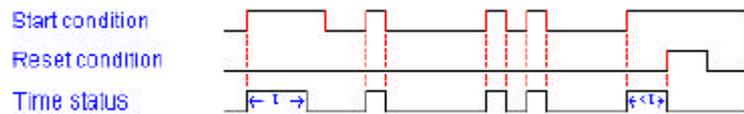


Figura 10. Demonstrativa de como o loop do Timer funciona.

### 3.1.7. Timer - Extended Pulse (SV)

Esse Timer depende para ser iniciado de um pulso prolongado também com mudança de nível lógico 1 para 0 como mostra a (12). Como o anterior o tempo do loop será definido após sua nomeação [4].



Figura 11. Simbologia do Timer.

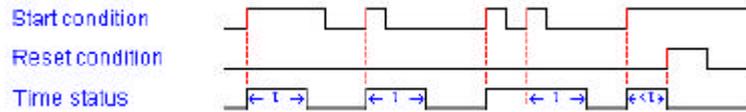


Figura 12. Demonstrativa de como o loop do Timer funciona.

### 3.1.8. Timer - Stop (TH)

Este elemento para o loop de um Timer. O nome do loop do Timer (por exemplo “T1”) é dado no campo da entrada acima do elemento do Timer. Anote por favor que a utilização do TH só tem sentido somente se um Timer tiver começado antes. Por exemplo, se você executar o T1 do laço do Timer como um pulso (SI), você pode para este laço do Timer através de uma outra filial do LD [4].



Figura 13. Simbologia do Timer de stop.

### 3.1.9. Timer - Reset (RT)

Este elemento zera a contagem de um Timer já iniciado [4].



Figura 14. Simbologia do Timer de reset.

### 3.1.10. Contador Set

Um contador é carregado com um valor nominal a ser contado. o nome do contador (por exemplo “C1”) é dado no campo acima do elemento. Em cima ao nome do contador em um outro campo aparecerá qual é o valor nominal do contador é incorporado [4], como mostra a (15).



Figura 15. Simbologia do contador.

Uma função muito interessante é a Check Branch, que verifica a atual programação para encontrar possíveis erros. Se nenhum erro for encontrado, a mensagem de “programação sem erros” aparecerá. Além disso, a programação é otimizada. Neste processo, todas as conexões entre elementos são encurtadas a seu melhor comprimento (veja (16) e (17)).



Figura 16. Antes da otimização.



Figura 17. Depois da otimização.

## 3.2. PROGRAMANDO O CLP

Para o desenvolvimento do projeto foi utilizado um CLP fabricado pela Bosch (CL-150), o qual utiliza o software WinSPS para sua programação. Esse CLP apresenta:

- 8 entradas digitais 24 V;
- 8 saídas digitais 24V / 0,5A;
- 2 contadores de 32 bits de alta velocidade;
- 3 entradas de alarme.

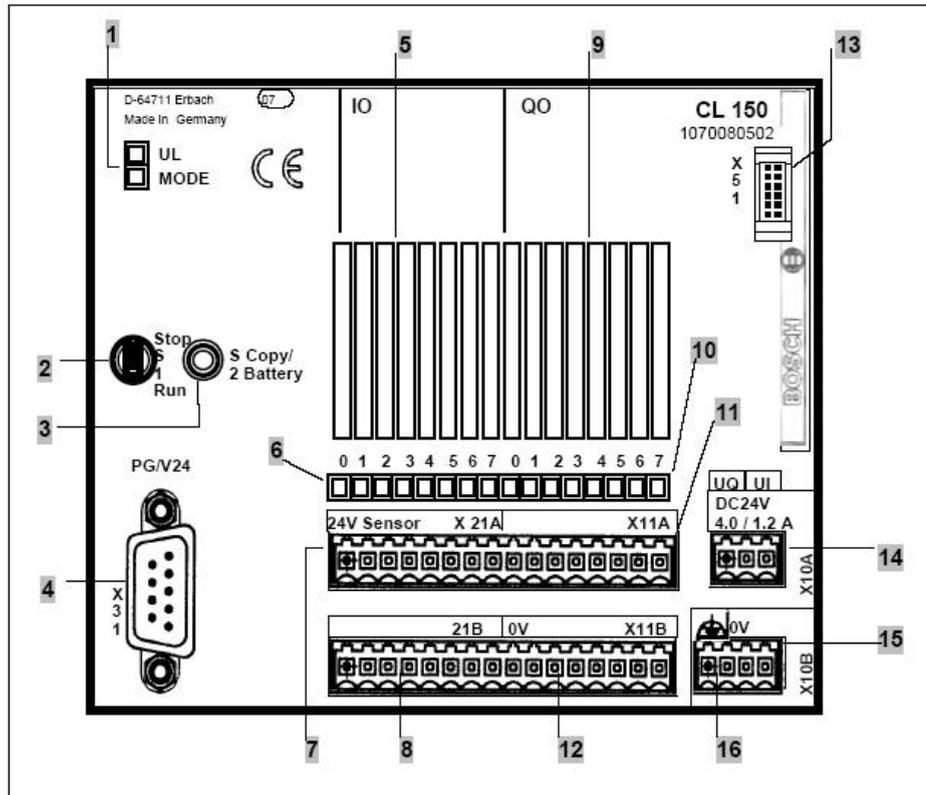


Figura 18. CL 150.

<b>1</b>	LED display
<b>2</b>	Toggle switch Stop/Run
<b>3</b>	Button Copy/Battery
<b>4</b>	V.24 interface for connecting programming units
<b>5</b>	Labeling field for digital inputs
<b>6</b>	Status display for digital inputs
<b>7</b>	24V outputs for sensor power
<b>8</b>	Digital inputs
<b>9</b>	Labeling field for digital outputs
<b>10</b>	Status display for digital outputs
<b>11</b>	Digital outputs
<b>12</b>	0V reference potential for actuators
<b>13</b>	Connector for the B-I/O modules
<b>14</b>	24V supply
<b>15</b>	0V reference potential for supply voltages
<b>16</b>	Functional earthing

Tabela 1. Especificação do CL 150.

Para executar a programação do CL-150, é necessário que os seguintes passos sejam seguidos.

### 3.2.1. 1º Passo

O CLP deve ser alimentado com 24V.

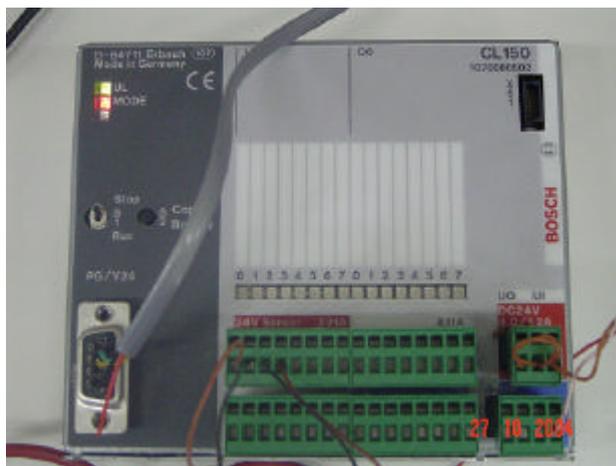


Figura 19. CLP alimentado pronto para ser usado.

### 3.2.2. 2º passo:

Depois de feita a rotina deve-se salvar o arquivo e verificar se existe comunicação entre o CLP e o computador. Essa comunicação é feita através da porta serial do computador e o conector X31 do CLP.



Figura 20. CLP conectado na porta serial.



Figura 21. Confirmação de comunicação.

### 3.2.3. 3º passo

Com a confirmação de comunicação entre os dois, colocar o CLP em modo STOP para carregar a rotina.



Figura 22. Chave STOP\RUN.

### 3.2.4. 4º passo

O próximo passo é mandar carregar o programa para o CLP.

### 3.2.5. 5º passo

Ao finalizar a troca de informações entre o CLP e o computador deve colocar o CLP em modo RUN. Caso seja necessário o software apresenta uma ferramenta que permite ao programador acompanhar a rotina em andamento.

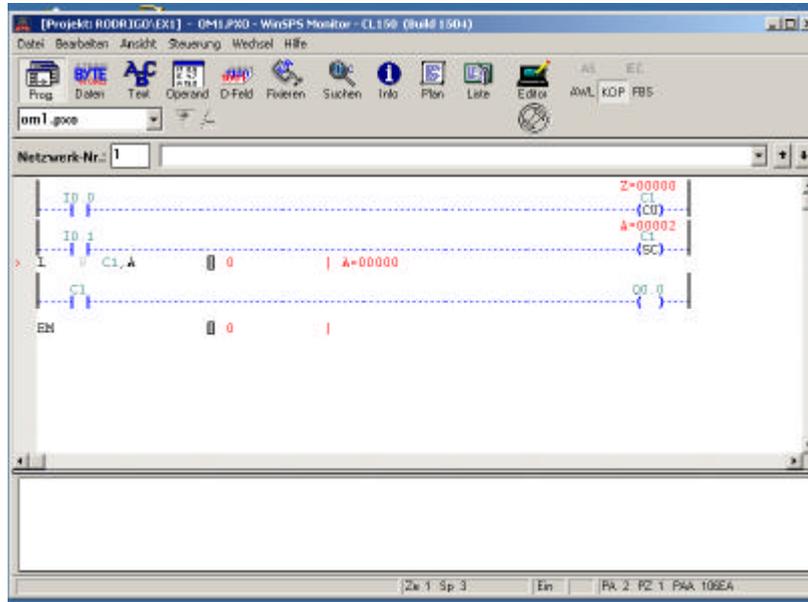


Figura 23. Acompanhamento simultâneo.

### 3.3. VALIDAÇÃO

A figura 24 mostra como é composto o posto de trabalho fisicamente.

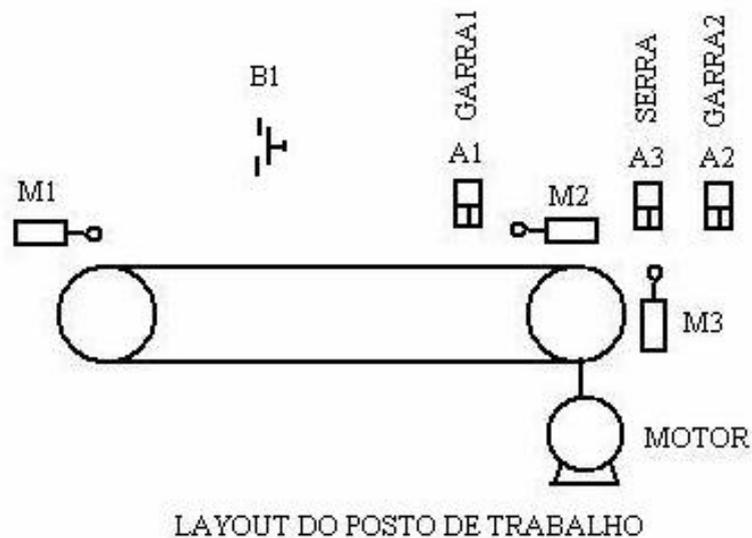


Figura 24. Layout do posto de trabalho.

A Figura 25 apresenta o modelo de funcionamento do posto de trabalho através da linguagem Grafset, mostrando a seqüência com que as ações serão tomadas. A seguir será explicado o funcionamento e a programação de cada etapa.

Abreviatura	Descrição	I/O do CLP
B1	Botão de Start	I0.0
M1	Chave fim de curso localizada no final da mesa	I1.0
M2	Chave fim de curso localizada no início da mesa	I2.0
M3	Chave fim de curso localizada abaixo da serra	I3.0
B2	Botão de emergência	I4.0
A1	Atuador Garra 1	Q0.0
A2	Atuador Garra 2	Q2.0
A3	Atuador da Serra	Q3.0

Tabela 2. Descrição das abreviaturas utilizadas no modelo.

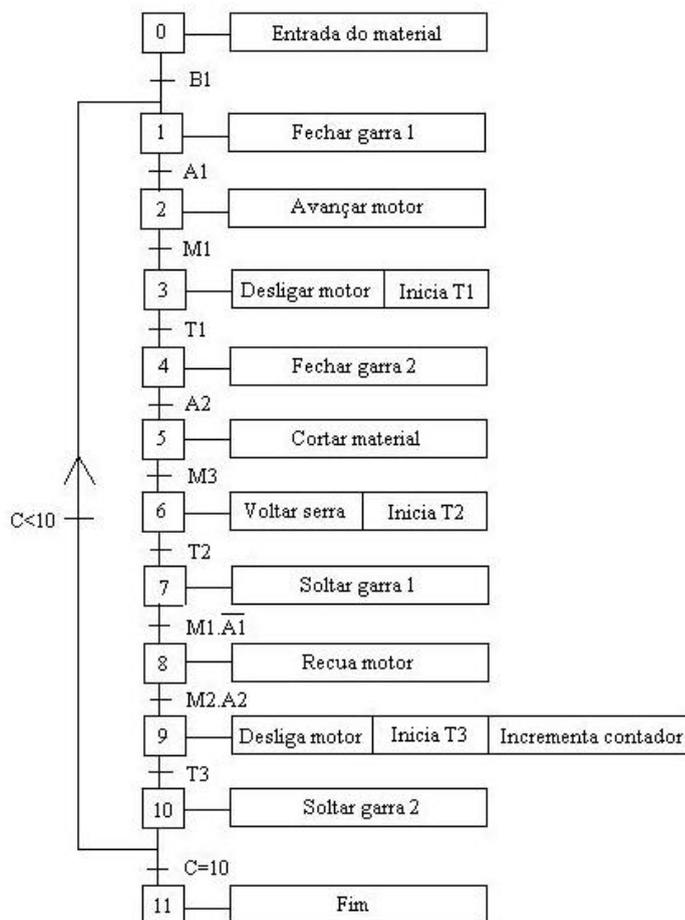


Figura 25. Modelo do posto de trabalho.

As explicações de cada etapa serão apresentadas através da sua programação correspondente no WinSPS e uma dissertação sobre a mesma.

### 3.3.1. Etapas 0 e 1



Figura 26. Etapas 0 e 1.

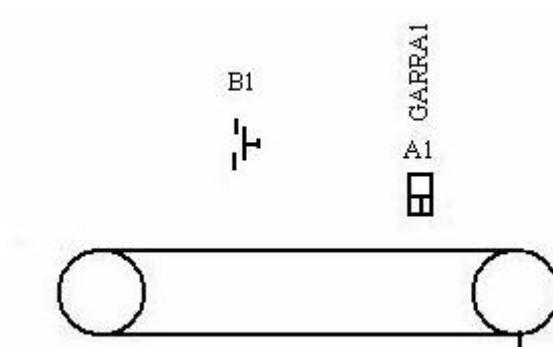


Figura 27. Elementos utilizados nas etapas 0 e 1.

Essas etapas serão responsáveis por iniciar o processo. Na etapa 0, o material será colocado no sistema para liberação do processo. A etapa 1 acionará a garra 1 que tem como função prender o material que será cortado, esse acionamento só acontecerá após o botão 1 (B1) ser apertado.

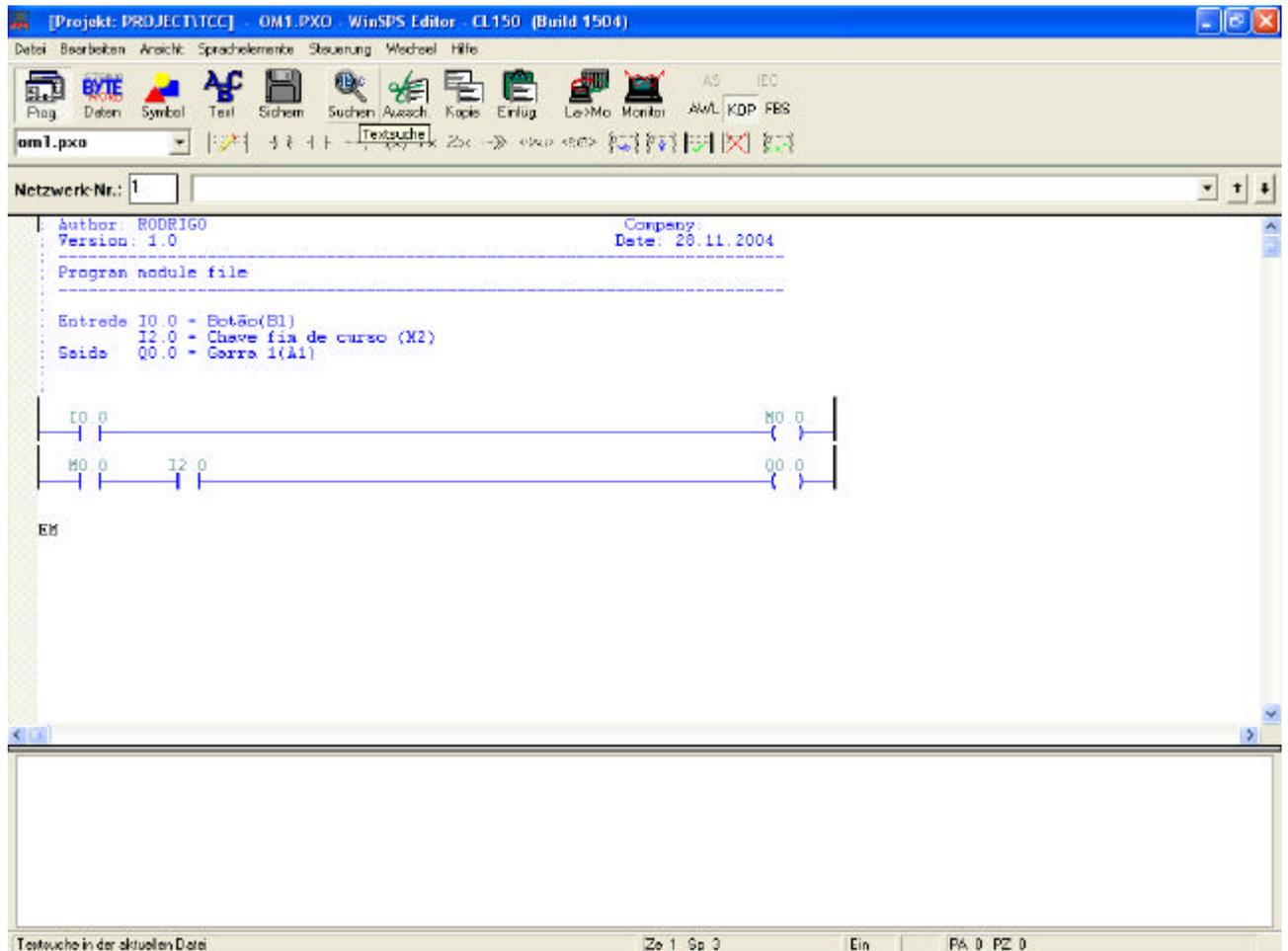


Figura 28. Programa relacionado à etapa 0 e 1.

### 3.3.2 Etapas 2 e 3

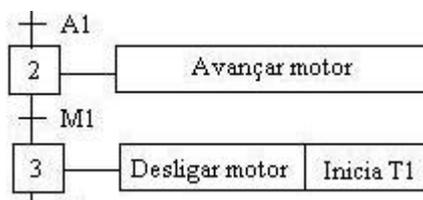


Figura 29. Etapas 2 e 3.

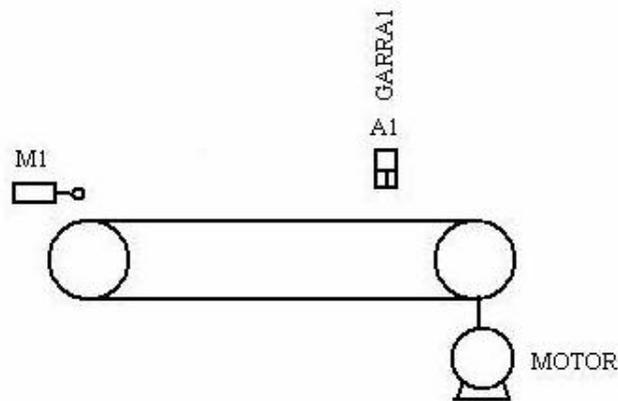


Figura 30. Elementos utilizados nas etapas 2 e 3.

Com o acionamento da garra 1, o motor será ligado de modo que seu sentido de rotação seja de levar o material até o ponto de corte (avança a garra). Mais a frente notará que o mesmo motor utilizado para levar o material até o ponto de corte, terá a função de trazer a garra 1 para seu ponto inicial; como o CLP utilizado não contém saída analógica, foi necessária a utilização de dois relês para comandar o sentido de rotação do motor.

O avanço do motor será interrompido quando a chave fim de curso M1 for acionada. Um temporizador T1 será disparado com a intenção de gerar uma segurança para o sistema não permitindo que a serra corte o material com o motor ainda em movimento.

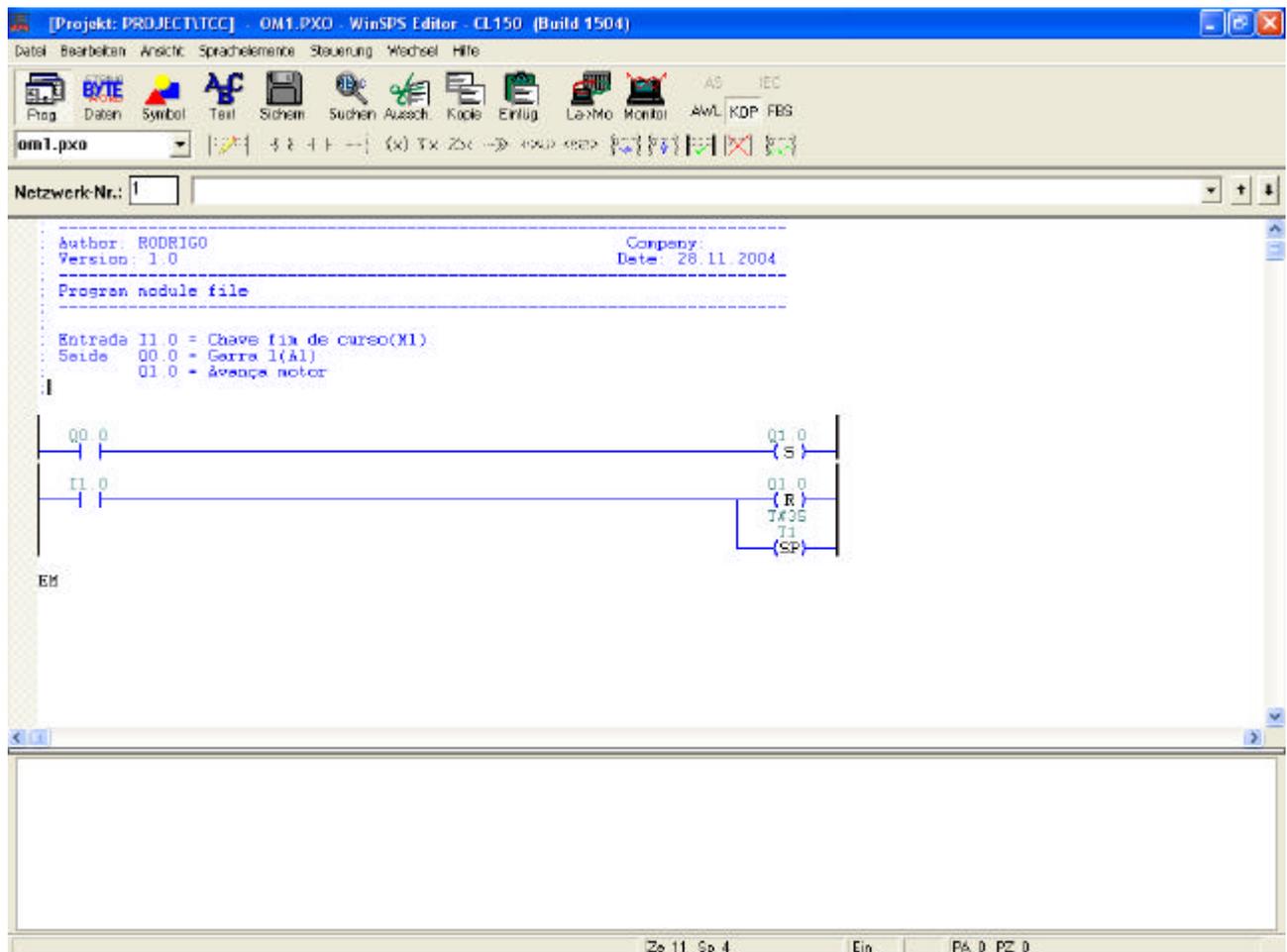


Figura 31. Programa relacionado à etapa 2 e 3.

### 3.3.3. Etapas 4, 5 e 6

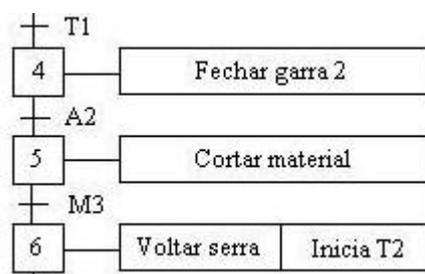


Figura 32. Etapas 4, 5 e 6.



### 3.3.4. Etapas 7, 8, 9 e 10

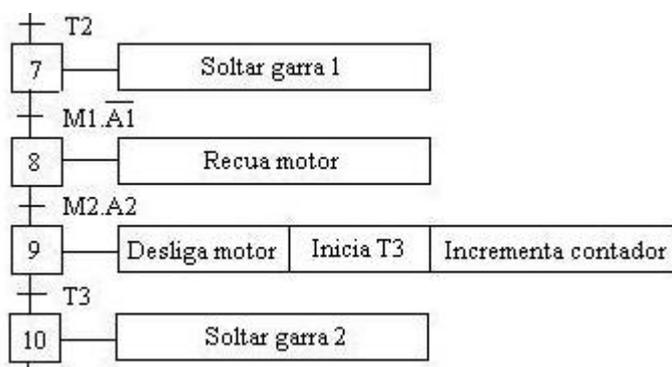


Figura 35. Etapas 7, 8, 9 e 10

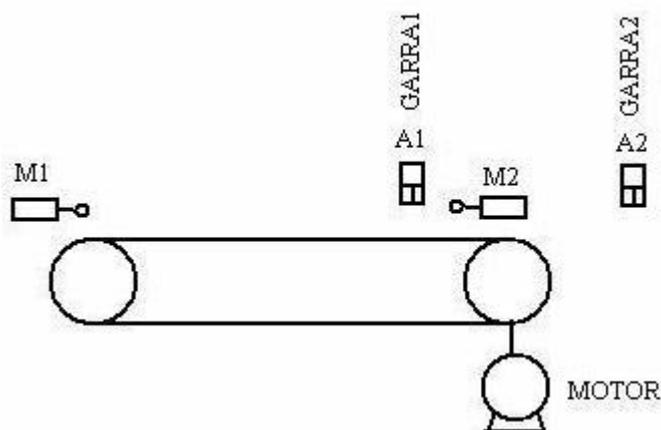


Figura 36. Elementos utilizados nas etapas 7, 8, 9 e 10.

Com a chave M1 acionada, e T2 igual a um, a garra 1 soltará o material, o motor será ligado no sentido contrario (recua a garra) e desligará com o acionamento da chave fim de curso M2 que se encontra no ponto inicial. O acionamento de M2 incrementará o contador e iniciará a contagem de T3. Quando T3 alcançar seu valor de setpoint, a garra 2 também se soltará.

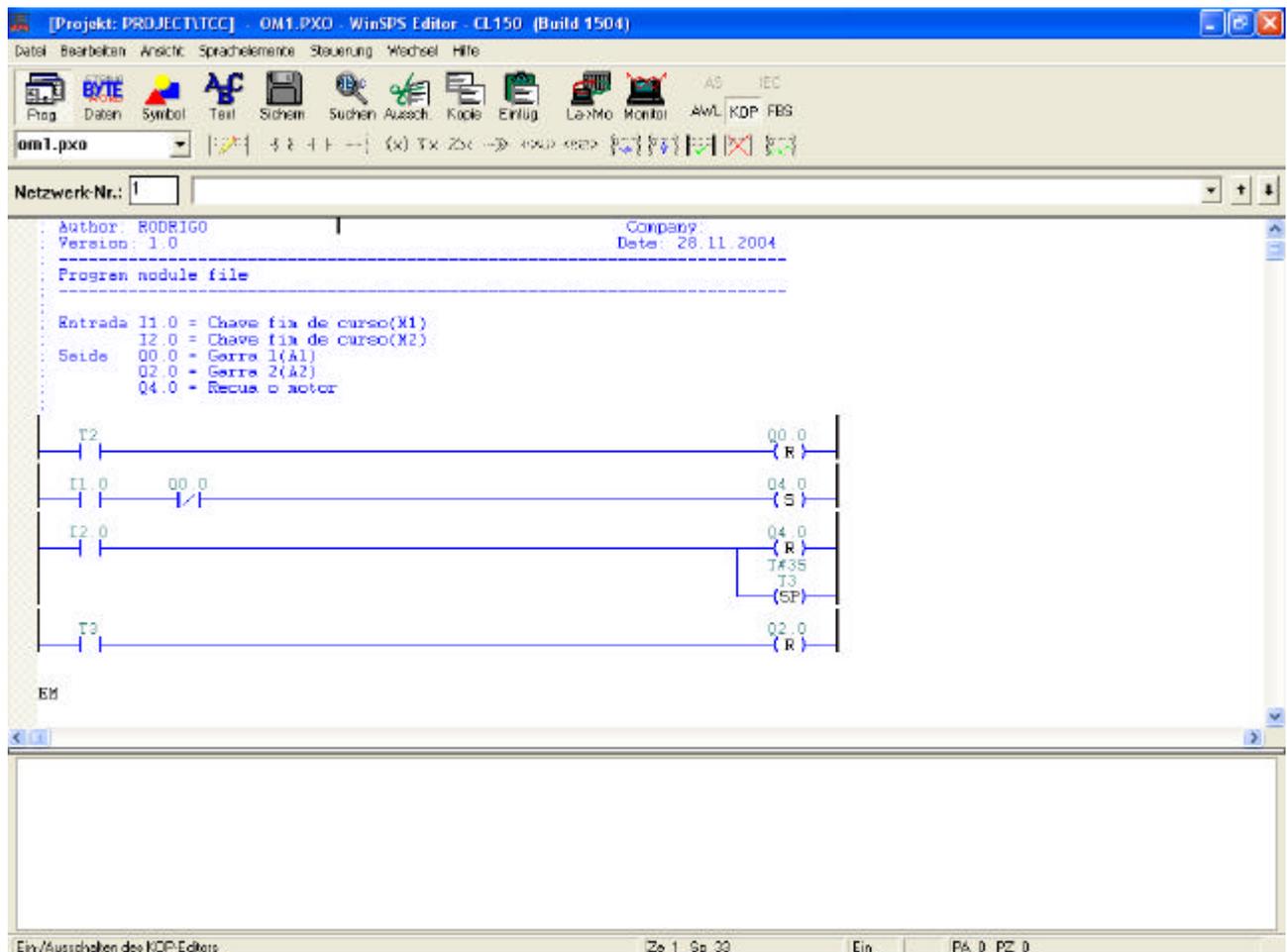


Figura 37. Programa relacionado à etapa 7, 8, 9 e 10.

A Figura 32. apresenta a estrutura do contador. Nesse caso a entrada I2.0 sempre que acionada irá gerar um pulso para o contador UP, esse valor será comparado com o setpoint através da variável M10.3 (variável de entrada específica para executar essa comparação). Quando M10.3 “enxergar” uma igualdade entre os valores, um sinal é enviado para saída Q5.0.

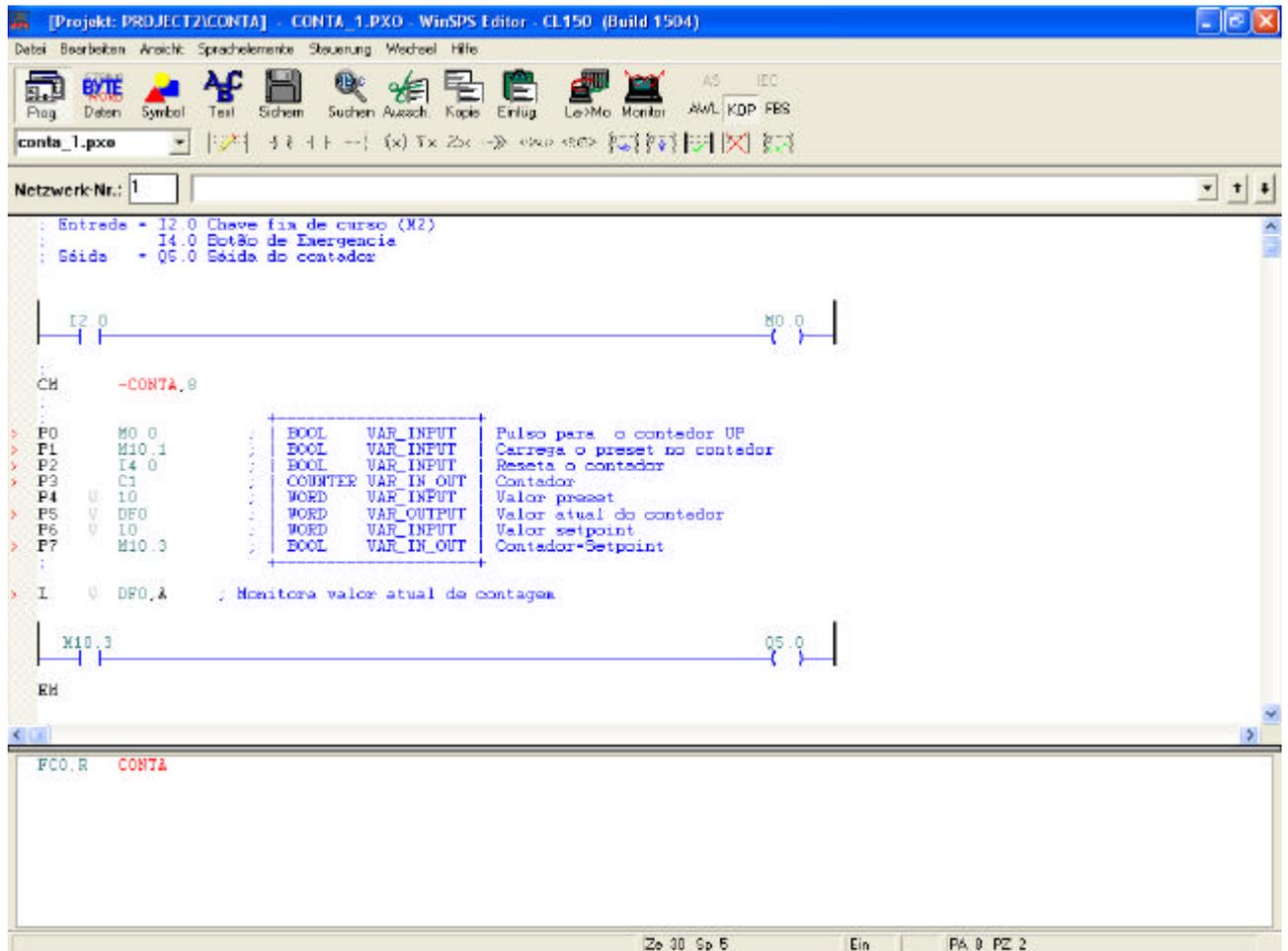


Figura 38. Programa do Contador.

### 3.3.5. Etapa 11

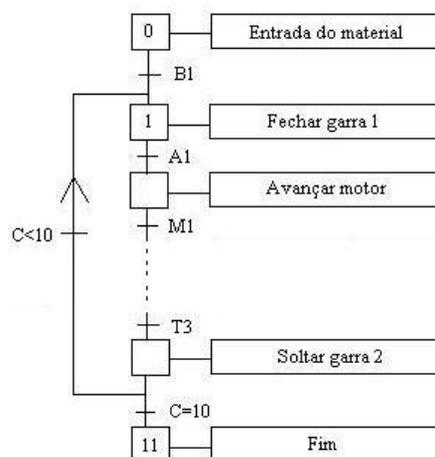


Figura 39. Etapa 11.

Após todo o processo de corte do produto, o programa irá comparar o valor atual do contador com o valor do setpoint caso esses valores sejam diferentes, o programa volta para a etapa 1 repetindo todo o processo novamente. Quando os valores se igualarem o programa será finalizado.

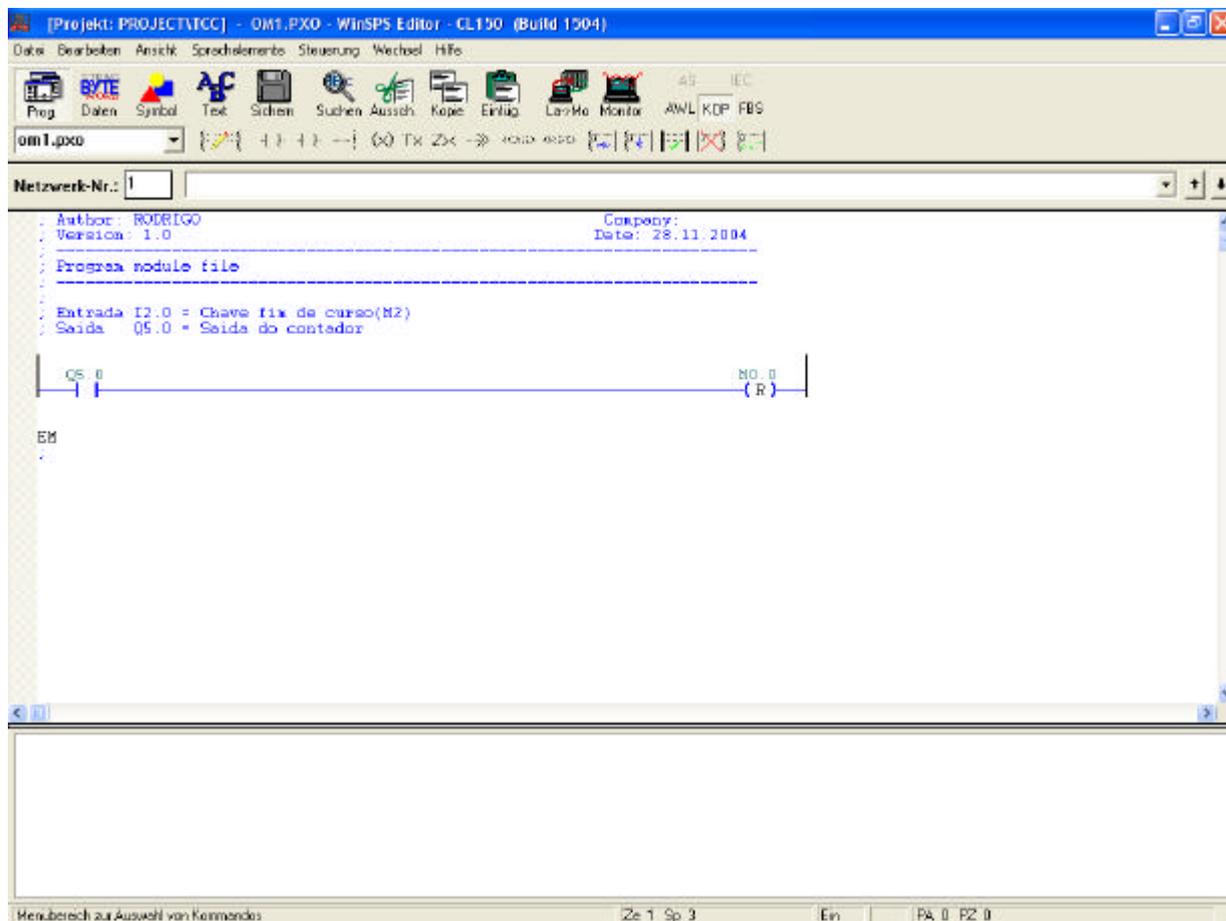


Figura 40. Programa que encerra o processo.

### 3.4. SIMULAÇÃO

Para a simulação do programa foi utilizado um motor de corrente contínua 12Vdc. Esse motor foi escolhido por proporcionar uma fácil alteração no sentido de rotação, basta que invertamos o sinal positivo e negativo nos terminais do motor. Para essa inversão dois relês 24V foram conectados em saídas distintas do CLP. O motivo de utilizar o relê de 24V é justamente pela tensão de saída do CLP ser de 24V assim não seria necessário a utilização de abaixadores de tensão simplificando a placa de interface entre motor e CLP.

O acionamento das chaves fim de curso, foi feito manualmente com a utilização de cabos presos às entradas do CLP. O recurso de monitoramento oferecido pelo software foi bastante

utilizado, visto que o mesmo acompanha todos os movimentos e acionamentos efetuados pelo CLP, inclusive os temporizadores e contadores como mostra (22). Abaixo algumas fotos tiradas na simulação.

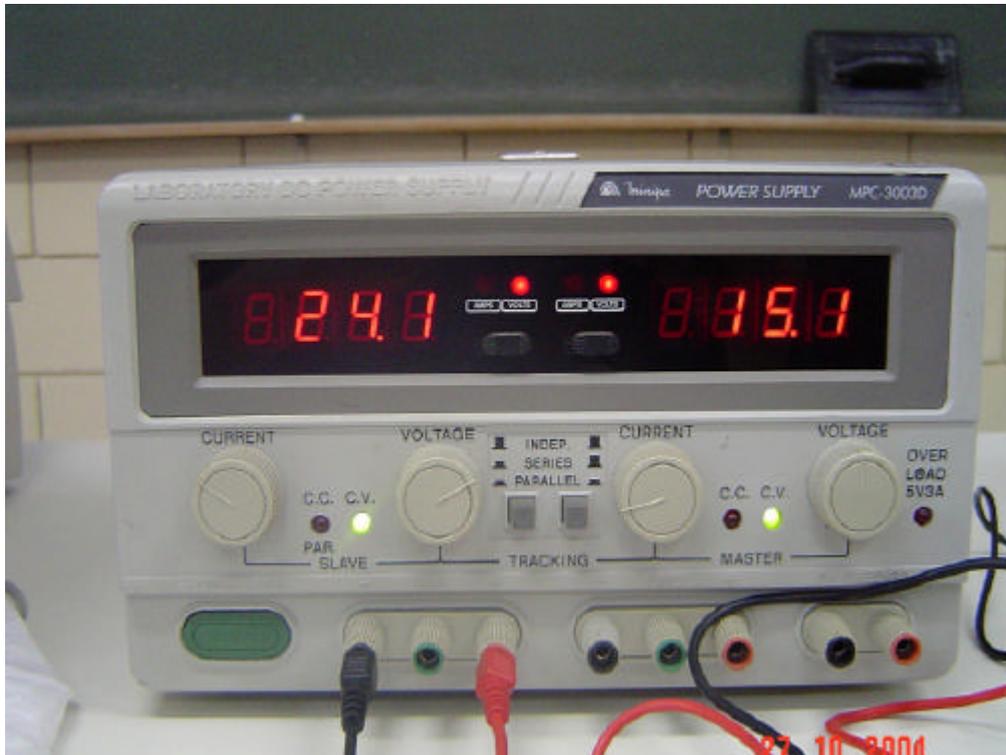


Figura 41. Fonte ajustada para alimentar o CLP.

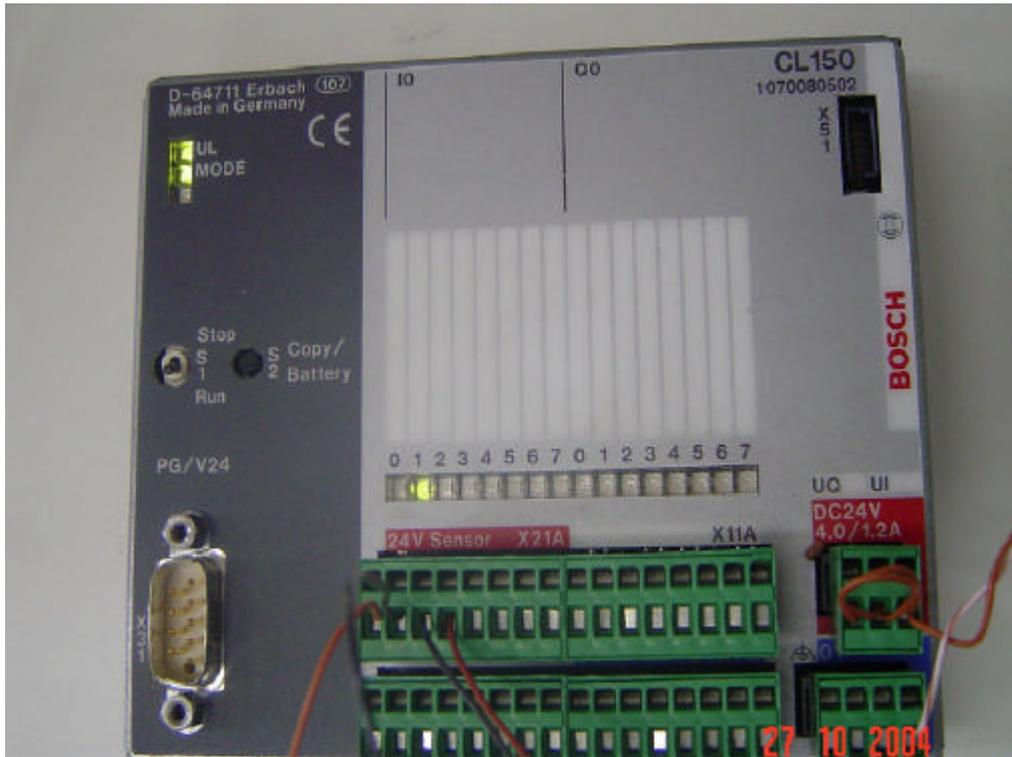


Figura 42. Entrada I1.0 sendo acionado.

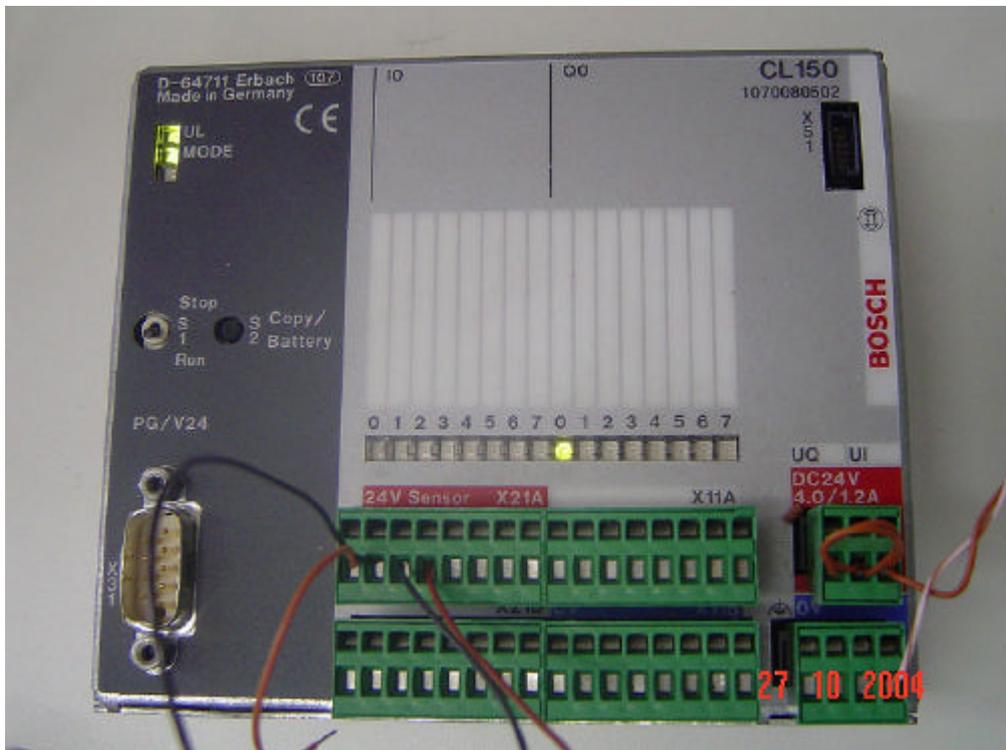


Figura 43. Saída Q0.0 ativada.

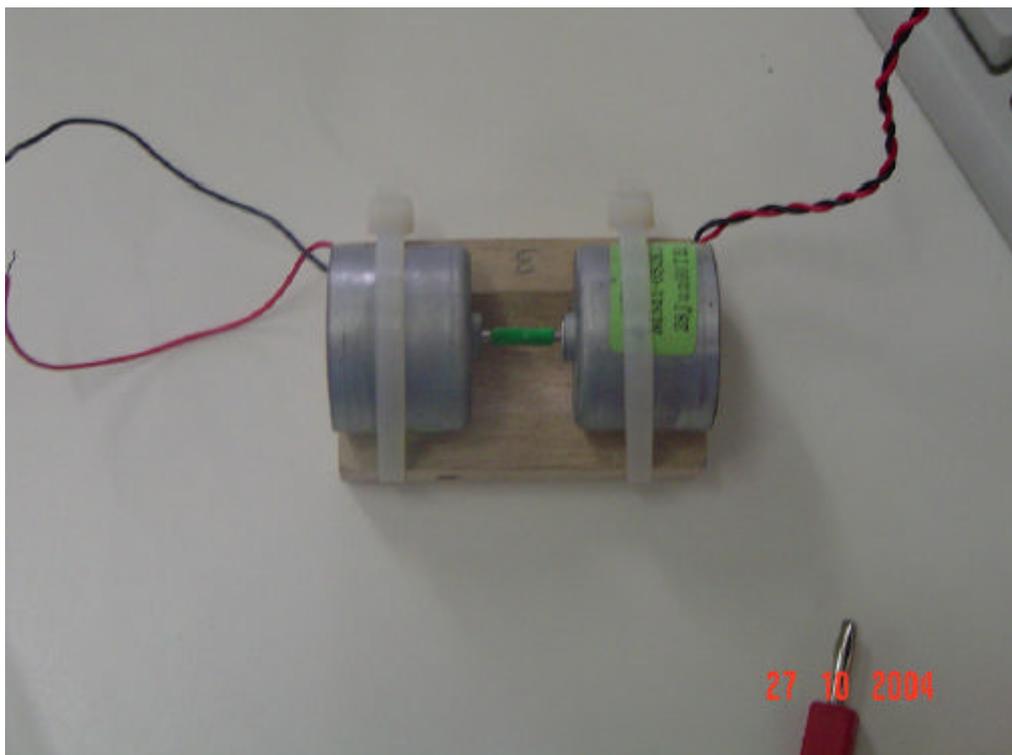


Figura 44. Motor utilizado na simulação.



Figura 45. Relê utilizado para controle do motor.

## **CONSIDERAÇÕES FINAIS**

Nesse trabalho foi apresentado a integração e o controle de um posto de trabalho. Os resultados finais obtidos foram satisfatórios, visto que nosso objetivo foi alcançado. Uma grande vantagem encontrada é a facilidade com que os componentes do posto de trabalho são integrados e controlados com a utilização de um CLP. Notamos isso quando partimos para a lógica do processo que é descrita em poucas linhas de programação.

Outro ponto importante é ressaltar o acerto na escolha do CLP a ser utilizado, que forneceu todos os recursos necessários para o desenvolvimento do trabalho. Gerando assim uma automatização bastante eficiente e confiável.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SILVEIRA, Paulo R.da; SANTOS, Winderson E., *Automação e controle discreto*, 1º ed., Editora Érica, 1999.
- [2] GEORGINI, Marcelo, *Automação Aplicada*, 1º ed., Editora Érica.
- [3] Manual técnico do CL150.
- [4] Manual do software WINSP.