

MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

***Interconexão Robô – CNC via controlador lógico programável***

**Carlos Dreyer Neto**

Orientador: Prof. *Dr. Eduardo A. Perondi*

Co-orientador: Prof. *Dr. Flávio J. Lorini*

Área de Concentração: Projeto e Fabricação

Resumo

O presente trabalho apresenta o processo de implementação de uma interconexão de dispositivos para a construção de uma célula de manufatura automatizada onde um robô é responsável pelo abastecimento de um CNC em um processo de usinagem. A comunicação Robô-CNC é estabelecida através de comandos lógicos binários programados de forma seqüencial em um CLP. O projeto do comando foi realizado através da aplicação combinada de diversas técnicas usualmente utilizadas nesta área. Os resultados obtidos demonstram que a solução proposta é capaz de controlar a célula em questão com a segurança desejada.

Abstract

"Interconnection Robot-CNC through Programmable Logical Controller"

This work presents the interconnection devices implementation process for construction of an automated manufactured cell where a robot is responsible for provisioning a CNC in a drilling process. The robot-CNC communication is established through binary logical commands programmed in a sequential form in a PLC. The project of the command was accomplished through the combined application of several techniques usually used in this area. The results obtained demonstrate that the solution proposed is able to control the cell in question with the expected safety.

PALAVRAS CHAVE

Célula de manufatura, projeto de comandos seqüenciais, robótica e fabricação automatizada.

## SUMÁRIO

1. INTRODUÇÃO.....	3
2. REVISÃO BIBLIOGRÁFICA.....	4
2.1 Redes de Chão de Fábrica.....	4
2.2 Controlador Lógico Programável - CLP.....	5
2.2.1 Características das Entradas.....	6
2.2.2 Características das Saídas.....	6
3. METODOLOGIA.....	6
4. EXPERIMENTOS DE CAMPO.....	16
5. RESULTADOS.....	18
6. CONCLUSÃO.....	18
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	19
8. ANEXOS.....	20
8.1 Anexo 1: Diagrama lógico.....	20
8.2 Anexo 2: Diagrama lógico seqüencial.....	21
8.3 Anexo 3: Programa utilizando linguagem do CLP Dexter.....	22

## 1. INTRODUÇÃO

Um mercado exigente e a concorrência cada vez mais acirrada tornam a necessidade de um processo de produção mais qualificado e eficiente um fator decisivo para a sobrevivência de uma empresa. A evolução do conceito do emprego de máquinas projetadas especificamente para produção em série, amplamente difundido por Henry Ford no início do século XX, juntamente com o avanço tecnológico nas áreas de mecânica, eletrônica digital e ciência da computação, possibilitaram a criação de sistemas integrados de automação industrial.

Devido à constante necessidade de evolução dos produtos fabricados e ao alto custo inicial da própria célula, a flexibilidade apresenta-se como a principal característica de uma célula automatizada. A flexibilidade, do ponto de vista econômico, é uma das chamadas *dimensões competitivas*, sendo um dos principais fatores levados em conta para o estudo do investimento.

Entende-se como “flexibilidade” a capacidade de a célula automatizada se ajustar rapidamente às variações da demanda ou modificações realizadas nos produtos, bem como, a capacidade de se adequar à produção de mais de um produto e também ser capaz de interagir com outras células a fim de possibilitar um melhor controle da produção (Black, 1991).

Os modernos dispositivos microprocessados permitem que cada componente presente no sistema seja capaz de interagir com os demais. A autonomia e a flexibilidade criadas para estes dispositivos permitem, segundo Romano (2002), a formação de um sistema de automação capaz de fabricar produtos de alta qualidade a custo e tempo reduzidos.

Um grande problema, entretanto, é tornar estes dispositivos compatíveis entre si, ou seja, criar uma interconexão de comunicação entre os elementos do sistema.

O presente trabalho tem como objetivo apresentar o estabelecimento de uma comunicação entre um robô ABB (IRB 1400) e um centro de usinagem CNC Romi (Discovery 306) instalados fisicamente no Laboratório de Robótica do GPFAI (Grupo de Projeto, Fabricação e Automação Industrial - DEMEC – EE - UFRGS) através de um controlador lógico programável Dexter ( $\mu$ DX<sup>+</sup> série 100), de forma que ambos possam interagir entre si, bem como com outros dispositivos de controle. Como resultado final deste trabalho pretende-se obter um sistema que permita estabelecer a comunicação da célula e que possa ser testado através de um exemplo de aplicação prática. Será, portanto, definido um procedimento de trabalho conjunto entre o robô e o CNC supervisionado pelo CLP. As tarefas executadas, apesar de relativamente simples, deverão ser comprobatórias do desempenho da solução adotada.

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1 REDES DE CHÃO DE FÁBRICA

Para a automação de um processo faz-se necessária a interação dos dispositivos de um sistema tanto entre si como com o processo em questão. As “interfaces” de comunicação têm como objetivo captar e transmitir informações provenientes de sensores e transdutores, bem como sinais de controle e atuação, fazendo com que a interligação dos dispositivos seja possível.

Com relação ao número de dispositivos interconectados, os modos de comunicação de um sistema de automação industrial podem ser classificados como *ponto a ponto* e *multiponto*. Na comunicação ponto a ponto um dispositivo comunica-se diretamente, por meio físico, com outro. Já em um sistema com comunicação multiponto um dispositivo pode se comunicar com diversos outros dispositivos.

A interligação de diversos dispositivos por um único meio físico conhecida como barramento serial ou paralelo é a topologia mais utilizada em chão de fábrica. Os barramentos industriais, também chamados de *fieldbuses*, apresentam diversas vantagens. Dentre elas pode-se citar a redução dos cabos utilizados que ocasionam a conseqüente diminuição de gastos com material e manutenção.

O uso de uma configuração onde diversos dispositivos são interligados através de meio físico compartilhado cria a necessidade de se definir um conjunto de regras e convenções que controlem a seqüência e o modo como as informações serão enviadas. Isto evita o congestionamento de informações e distribui as mesmas de uma forma mais rápida. Esta padronização é obtida com a criação de protocolos normalizados de comunicação, sendo o *Profibus* e *CAN* os mais utilizados atualmente.

O protocolo PROFIBUS (*PROcess FIEldBUS*) é o sistema de barramento mais utilizado na Europa e está em crescente aceitação pelo resto do mundo (PROFIBUS, 2002). O acesso do dispositivo ao meio físico pode ser realizado por dois métodos: comunicação mestre/escravo e anel lógico de passagem de *token* (permissão de comunicação) entre os mestres.

No primeiro caso, toda comunicação é inicializada por um mestre, enquanto que os escravos apenas respondem às requisições. Esta configuração torna o escravo um elemento passivo, fazendo com que o início de uma comunicação nunca parta de um escravo.

O anel lógico *token* é uma configuração na qual a permissão para comunicar transita entre os mestres, só podendo comunicar o mestre que estiver com a permissão.

O CAN (Computer Area Network) é um protocolo através do qual qualquer dispositivo

pode iniciar a comunicação (multimestre), e utiliza uma estratégia de acesso múltiplo com contenção para evitar colisões.

## 2.2 CONTROLADOR LÓGICO PROGRAMÁVEL – CLP

Os controladores lógicos programáveis (CLPs) surgiram para substituir os antigos painéis de controle a relé que necessitavam de modificações na fiação para sua reprogramação (Richter, 2001). Os CLPs, por sua vez, são de fácil reprogramação, pois esta se dá no *software* e não no *hardware*, como ocorre nos painéis.

O advento dos microprocessadores aumentou o poder de processamento e confiabilidade dos CLPs, possibilitando o surgimento das redes locais para comunicação de CLPs entre si, bem como com computadores.

O controlador programável utilizado neste trabalho,  $\mu$ DX Série 100 Plus, possui 4 entradas, 4 saídas e *software* de programação (programa PG). Este *software* facilita a programação do  $\mu$ DX, pois é baseado em ícones (blocos de funções), ligados por fios (nodos), desenhados em tela gráfica colorida, com utilização do *mouse* para posicionamento.

O controlador permite conexão a outros 14 controladores, via rede DXNET. A rede local é multi-mestre com topologia tipo barramento.

O controlador  $\mu$ DX Plus possui também a informação de Conjunto DXNET. Cada conjunto (sendo possível haver até 16 conjuntos) pode ser formado por até 16 controladores  $\mu$ DX Plus, perfazendo 256 endereços DXNET diferentes. Assim, no caso de comunicação via rádio-transmissão ou via rede RS-485 podem ser ligados até 256 controladores a uma estação central.

O programa é gravado no CLP em memória não-volátil (EEPROM). Além disso, possui previsão para pilhas comuns ou recarregáveis para manter a execução do programa durante até sete dias sem alimentação externa.

O controlador está preparado para suportar ambientes industriais, com ruídos elétricos e temperatura alta, por ter proteção contra transientes elétricos e ser acondicionado em gabinete metálico cujas dimensões são 117 x 100 x 33 mm. Sua temperatura de operação vai de 0°C até 60°C e é alimentado com energia elétrica de 9-14Vdc, 250mA (máximo) ou 4 pilhas tipo AA (ou baterias recarregáveis). Sua entrada E2 permite contagem rápida (até 3000 Hz, ou 180000 rpm) e a conexão de até 4 expansões de Entrada/Saída, perfazendo 36 entradas e 36 saídas (72 I/Os).

### 2.2.1 Características das Entradas

- Detecção de 1 lógico: +2,0 até +48,0V.
- Detecção de 0 lógico: -48,0V até +0,9V.
- Tensão máxima na entrada (1 minuto): 120VCA.
- Frequência de entrada: DC - 60Hz (260KHz para bloco PWMin).
- Tensão em +V: 9-14Vdc (conforme alimentação elétrica do  $\mu$ DX).

### 2.2.2 Características das Saídas

- Tensão máxima nos contatos: 30 VDC ou 220VCA.
- Corrente máxima: 10 A.
- Vida útil sem carga: 100.000.000 operações.
- Vida útil com carga resistiva máxima (CA): 80.000 operações.
- Isolação (entre bobinas e contatos): 2.000Vef (1 minuto).
- Tipo de saída: relé com contatos NA e NF disponíveis.

## 3. METODOLOGIA

A principal proposta deste trabalho é a criação de uma interconexão de comunicação de uma única célula. Assim, não há necessidade, em um primeiro momento, de se adotar um procedimento normalizado para a comunicação entre os dispositivos, uma vez que, devido à simplicidade do sistema em questão, o controle pode ser feito através de lógica sequencial simples. Porém, no caso de esta célula vir a ser integrada a outras, haverá a necessidade de estabelecer um padrão de comunicação para que a integração entre diversas células possa ser realizada com a flexibilidade necessária.

Definiu-se como tarefa a ser executada pelo sistema, a título de exemplo, um processo de usinagem no qual o Robô é o responsável pela alimentação de peças do CNC.

É importante destacar que a célula de trabalho é constituída por um CNC e um Robô, sendo a comunicação entre eles controlada por um CLP.

A primeira restrição para o controle do processo a ser implementado é o fato do CLP disponível possuir apenas quatro entradas e quatro saídas, o que cria a necessidade de nova simplificação.

O processo proposto inicia em uma condição onde o CNC está em estado de espera como, por exemplo, no término da usinagem de uma peça, sendo esta informação enviada para o CLP. Este por sua vez aciona o atuador responsável pela abertura da porta do centro de usinagem e informa ao robô que a mesma já está aberta, possibilitando que ele troque a peça. O robô, então, informa ao CLP a conclusão da sua tarefa e o controlador aciona o fechamento da porta do CNC, possibilitando que o CNC usine uma nova peça.

Devido à ausência de um sistema automático de abertura da porta do CNC, considera-se que o mesmo seja provido de um atuador pneumático com retorno automático através de mola e dois sensores fim de curso, um para porta aberta e outro para porta fechada.

A escolha de um atuador com retorno automático simplifica o sistema uma vez que, neste caso, é necessário apenas um sinal de saída do CLP para comandar a abertura e o fechamento da porta do CNC.

De posse destas especificações pode-se definir as variáveis de entrada e saída necessárias para o controle da célula, conforme mostrado na Tabela 3.1.

Tabela 3.1 - Variáveis de controle do sistema.

<b>Variáveis de Entrada</b>	<b>Notação</b>	<b>Correspondência Lógica</b>
E1	Sinal CNC	1
E2	Porta Aberta	1
E3	Robô no Set Point	1
E4	Porta Fechada	1
<b>Variáveis de Saída</b>	<b>Notação</b>	<b>Correspondência Lógica</b>
S1	Atuador (abrir)	1
S2	Robô Atuando	1
S3	Nova Tarefa CNC	1

O passo seguinte para a criação do comando seqüencial é a montagem de um Diagrama de Estados. Para a construção do mesmo, discrimina-se cada estado da seqüência, definindo-se as transições associadas às trocas de estados.

O Diagrama de Estados, elaborado para descrever o sistema em questão, pode ser observado na Fig. 3.1.

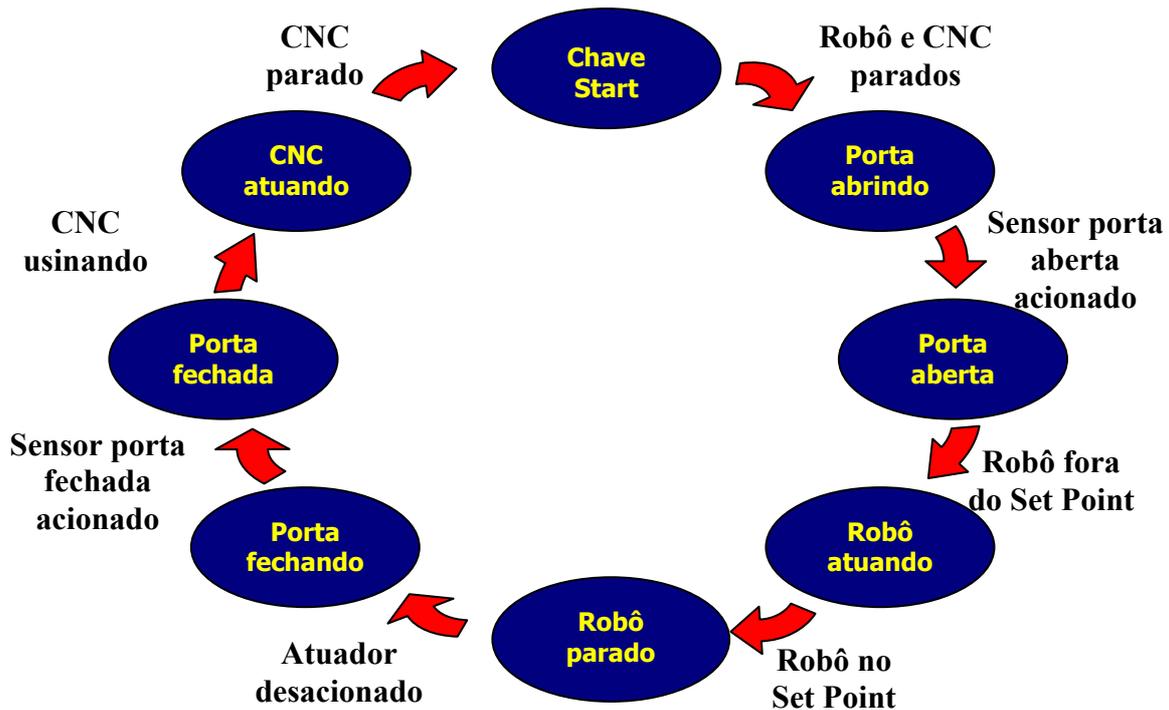


Figura 3.1 - Diagrama de Estados

O estado inicial é caracterizado pela alimentação do CLP acionada por meio de chave. Neste momento o CLP deve estar recebendo o sinal de que o robô está parado no *set point*, o sinal do CNC de que a peça já foi usinada e o sinal de porta fechada. Esta combinação de sinais, característica deste estado, faz com que o comando envie um sinal para o atuador abrir a porta do CNC.

Neste instante a seqüência entra em um estado transitório, quando a porta está entreaberta, ou seja, nenhum dos sensores de fim de curso está acionado, devendo, portanto, ser mantido o sinal de abertura da porta acionado no estado 1.

A condição de troca para o estado 2 é o recebimento do sinal de que a porta está aberta. Neste momento aciona-se um sinal para que o robô realize a troca da peça.

O sinal que indica que o robô está atuando é a condição de troca para o estado 3.

O recebimento do sinal de que o robô retornou ao *set point* é a condição para a troca de estado. No estado 4, a desativação do sinal para o atuador corresponde ao comando de fechamento da porta.

O sinal de porta fechada é a condição para a troca de estado. Neste momento existe novamente um estado intermediário, mas com todas as saídas desligadas.

No estado 6 é acionado o sinal para que o CNC usine nova peça. O não recebimento do sinal do CNC indica que o CNC está usinando e caracteriza o estado 7.

O recebimento do sinal de que o CNC está pronto é a condição para a transição para o novo estado, onde inicia-se um novo ciclo da seqüência.

Com base nestes dados, monta-se uma tabela na qual são colocadas todas as combinações possíveis entre os sinais entrada e as variáveis, relacionado-as com o estado lógico resultante das mesmas. Chama-se esta tabela de “Tabela Verdade”, a qual, para o presente problema, está apresentada na Tabela 3.2.

Tabela 3.2 - Tabela-Verdade

<b>Estado</b>	<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>S1</b>	<b>S2</b>	<b>S3</b>
<b>0</b>	1	0	1	1	1	0	0
<b>1</b>	1	0	1	0	1	0	0
<b>2</b>	1	1	1	0	1	1	0
<b>3</b>	1	1	0	0	1	0	0
<b>4</b>	1	1	1	0	0	0	0
<b>5</b>	1	0	1	0	0	0	0
<b>6</b>	1	0	1	1	0	0	1
<b>7</b>	0	0	1	1	0	0	0

Um método convencionalmente utilizado para a programação de controladores é o mapa de Karnaugh-Veitch que possibilita através de um trabalho sistemático a obtenção da equação lógica simplificada do controle do sistema em questão (Bollmann, 1996).

Este mapa é, na realidade, a própria Tabela-Verdade em uma forma matricial e contém todas as combinações dos sinais de entrada possíveis para cada saída. Na representação adotada  $\bar{E}$  significa que a entrada em questão tem valor igual a zero. Nas combinações que não correspondem a um dos estados, atribui-se o valor respectivo ao mesmo. Nos demais casos atribui-se o valor zero à saída.

Através da análise da tabela verdade, observa-se que existem estados com combinações das variáveis de entrada iguais, mas com saídas diferentes, ou seja, as saídas não são funções lógicas unívocas dos sinais de entrada. Devido ao fato de não poder-se utilizar outro sinal de controle, restrição imposta pelo CLP, deve-se, neste caso, realimentar-se dois sinais de entrada de forma a utilizá-los para a caracterização unívoca dos estados distintos entre si.

A realimentação dos sinais E1 e E2, aqui denominados de E1a e E2a, nos permite diferenciar todos estados entre si, resultando na nova Tabela Verdade (Tabela 3.3). A utilização de dois sinais realimentados caracterizou a criação de mais duas variáveis ao sistema, fazendo com que o mapa de Karnaugh-Veitch passasse a apresentar 64 termos, uma vez que existem  $2^n$  combinações possíveis, sendo  $n$  o número de variáveis.

Tabela 3.3 - Nova Tabela Verdade

Estado	E1	E2	E3	E4	E1a	E2a	S1	S2	S3
0	1	0	1	1	0	0	1	0	0
1	1	0	1	0	1	0	1	0	0
2	1	1	1	0	1	0	1	1	0
3	1	1	0	0	1	1	1	0	0
4	1	1	1	0	1	1	0	0	0
5	1	0	1	0	1	1	0	0	0
6	1	0	1	1	1	0	0	0	1
7	0	0	1	1	1	0	0	0	0

Existem programas computacionais que fornecem diretamente os algoritmos de controle através da tabela verdade. Porém, as versões de divulgação disponíveis estão restritas a no máximo quatro variáveis de entrada, impossibilitando sua aplicação no problema.

Assim, de posse destes dados, montaram-se manualmente os mapas de Karnaugh-Veitch para cada saída, conforme as figuras 3.2, 3.3 e 3.4.

E1, E2, E3, E4, E1a e E2a			$\bar{E}1a$				E1a			
			$\bar{E}3$		E3		E3		$\bar{E}3$	
			$\bar{E}1$	E1	E1	$\bar{E}1$	$\bar{E}1$	E1	E1	$\bar{E}1$
$\bar{E}2a$	$\bar{E}4$	$\bar{E}2$	0	0	0	0	0	1	0	0
		E2	0	0	0	0	0	1	0	0
	E4	$\bar{E}2$	0	0	1	0	0	0	0	0
		E2	0	0	0	0	0	0	0	0
E2a	E4	E2	0	0	0	0	0	0	0	0
		$\bar{E}2$	0	0	0	0	0	0	0	0
	$\bar{E}4$	E2	0	0	0	0	0	0	1	0
		$\bar{E}2$	0	0	0	0	0	0	0	0

Figura 3.2 - Mapa de Karnaugh-Veitch de S1.

E1, E2, E3, E4, E1a e E2a			$\bar{E}1a$				E1a			
			$\bar{E}3$		E3		E3		$\bar{E}3$	
			$\bar{E}1$	E1	E1	$\bar{E}1$	$\bar{E}1$	E1	E1	$\bar{E}1$
$\bar{E}2a$	$\bar{E}4$	$\bar{E}2$	0	0	0	0	0	0	0	0
		E2	0	0	0	0	0	1	0	0
	E4	$\bar{E}2$	0	0	0	0	0	0	0	0
		E2	0	0	0	0	0	0	0	0
E2a	E4	E2	0	0	0	0	0	0	0	0
		$\bar{E}2$	0	0	0	0	0	0	0	0
	$\bar{E}4$	E2	0	0	0	0	0	0	0	0
		$\bar{E}2$	0	0	0	0	0	0	0	0

Figura 3.3 - Mapa de Karnaugh-Veitch de S2.

E1, E2, E3, E4, E1a e E2a			$\bar{E}1a$				E1a			
			$\bar{E}3$		E3		E3		$\bar{E}3$	
			$\bar{E}1$	E1	E1	$\bar{E}1$	$\bar{E}1$	E1	E1	$\bar{E}1$
$\bar{E}2a$	$\bar{E}4$	$\bar{E}2$	0	0	0	0	0	0	0	0
		E2	0	0	0	0	0	0	0	0
	E4	$\bar{E}2$	0	0	1	0	0	0	0	0
		E2	0	0	0	0	0	0	0	0
E2a	E4	E2	0	0	0	0	0	0	0	0
		$\bar{E}2$	0	0	0	0	0	0	0	0
	$\bar{E}4$	E2	0	0	0	0	0	0	0	0
		$\bar{E}2$	0	0	0	0	0	0	0	0

Figura 3.4 - Mapa de Karnaugh-Veitch de S3.

Para a obtenção do comando simplificado deve-se agrupar o maior número possível de campos adjacentes que contém valor 1, sendo que estes devem ser termos que só se diferenciam por uma variável. Além disso, segundo Bollmann (1986), deve-se levar em conta que o número de campos adjacentes deve ser uma potência de 2 (ou seja,  $2^n$ ), e que a forma geométrica do agrupamento deve ser retangular.

Obtêm-se, assim, as seguintes funções para as saídas do sistema:

$$S1 = E1 * E3 * \bar{E}4 * E1a * \bar{E}2a * (E2 + \bar{E}2) + E1 * \bar{E}2 * E3 * E4 * \bar{E}1a * \bar{E}2a + E1 * E2 * \bar{E}3 * \bar{E}4 * E1a * E2a \quad (1)$$

$$S2 = E1 * E2 * E3 * \bar{E}4 * E1a * \bar{E}2a \quad (2)$$

$$S3 = E1 * \bar{E}2 * E3 * E4 * \bar{E}1a * \bar{E}2a \quad (3)$$

De posse das equações 1, 2 e 3, montou-se, através de operações booleanas o fluxograma lógico que as representa. Estas expressões podem ser simplificadas, obtendo-se circuitos lógicos mais compactos, com menor número de componentes.

Nota-se que o diagrama lógico obtido, Anexo 1, dificulta o entendimento do funcionamento do sistema e, conseqüentemente, aumenta a dificuldade de detecção de possíveis erros. O mesmo possui uma configuração difícil de programar no CLP.

Além disso, nesta configuração, o algoritmo não consegue reproduzir a seqüência desejada, pois a realimentação dos sinais através de um bloco pronto de memória nos fornece o sinal em um passo anterior, sendo, na verdade, necessário o uso do sinal em um estado anterior. Para se conseguir esta implementação deve-se, então, utilizar memórias conhecidas como memórias RS (ou *Set-Reset*).

Ao se projetar a nova configuração do fluxograma lógico (veja Anexo 2), observou-se que a lógica do problema envolve não apenas realimentação simples, mas um sequenciamento

com intertravamento das memórias RS. Montou-se, assim, um novo modelo, no qual a memória do estado inicial seria acionada através da combinação dos seus respectivos sinais de entrada com um pulso inicial ou, no caso do reinício da seqüência, com o sinal de acionamento do último estado. Este sinal aciona as ações do mesmo e serve como condição para a configuração do próximo estado. A combinação de sinais de entrada, junto com o sinal de saída da memória do estado anterior, garante que a execução da seqüência seja obedecida mesmo que existam estados com combinações de sinais iguais, conforme pode ser observado na Fig. 3.5. Assim, uma vez configurado determinado estado, uma das ações do mesmo deve ser o desacionamento do estado anterior.

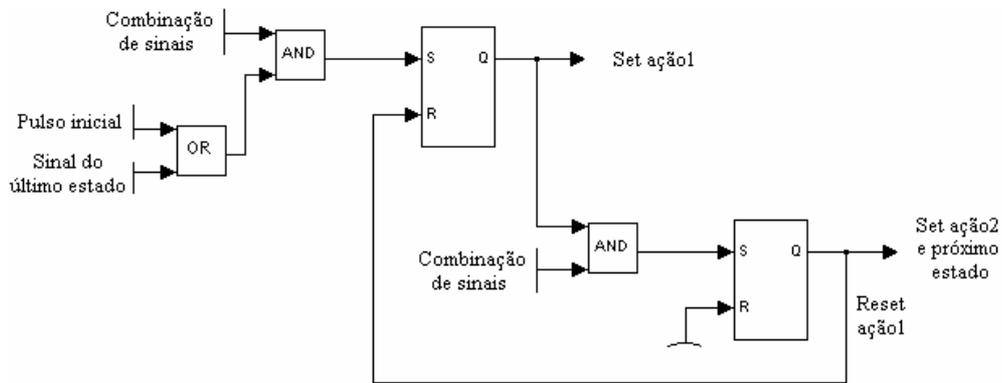


Figura 3.5 - Comando passo a passo seqüencial

Com base nestes dados, e com auxílio do Diagrama Trajeto-Passo (Fig. 3.6) conseguiu-se simular a seqüência de programação no programa de análise numérica e simbólica Simulink (Simulink, 1994), obtendo-se a seqüência desejada.

O Diagrama Trajeto Passo é a representação gráfica de eventos seqüenciais, nele podemos contemplar em todas as etapas do comando o estado dos dispositivos contidos no sistema (Bollmann, 1996).

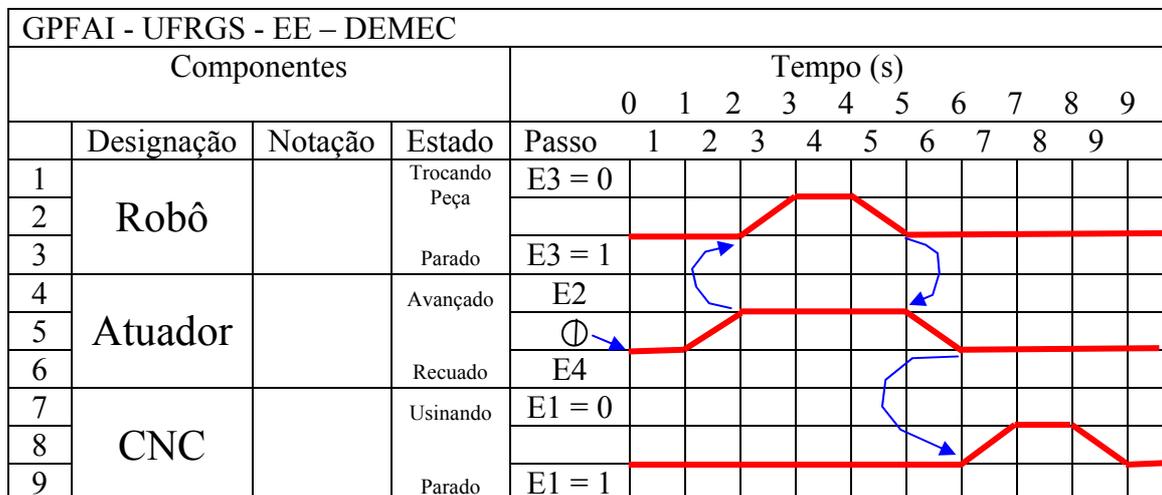


Figura 3.6 - Diagrama Trajeto-Passo.

Este novo modelo torna o fluxograma muito mais simples e sua montagem pode ser melhor entendida através do Gráfico Funcional de Comandos Etapa-Transição, ou Grafcet, que pode ser observado na Fig. 3.7. O Grafcet é constituído por uma simbologia gráfica com arcos orientados que interligam etapas por interpretação de variáveis de entrada e saídas (Silveira & Santos,1998). Esta configuração define de uma forma mais concreta, para uma seqüência, as condições de troca de estados, ou etapas, e suas ações.

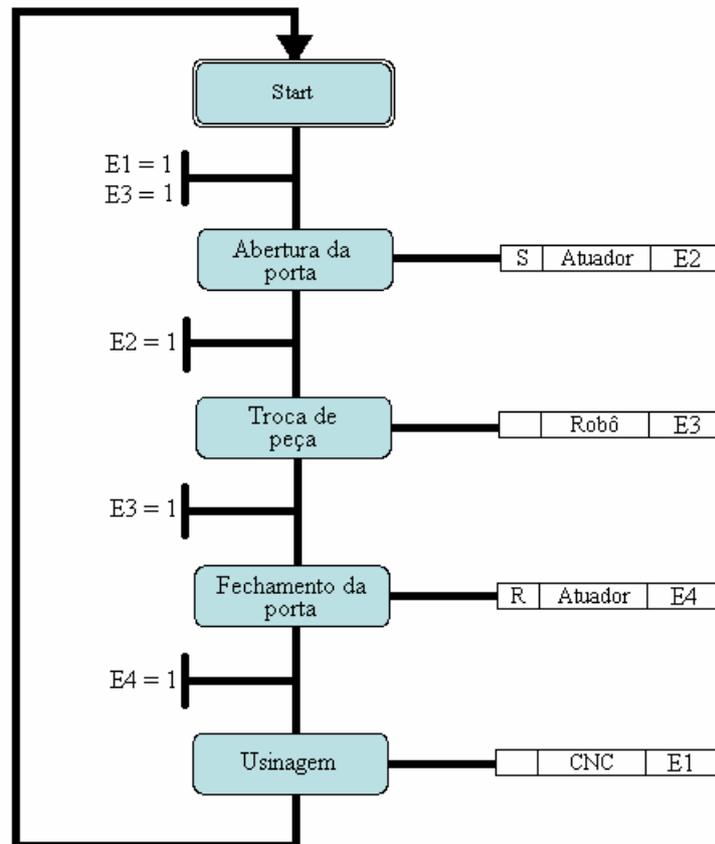


Figura 3.7 - Diagrama Grafcet

O diagrama lógico montado pode ser observado no Anexo 2. Vale ressaltar que o mesmo pode ser programado no CLP ou montado com componentes eletrônicos vendidos comercialmente.

Para a implementação desta configuração no CLP, deve-se utilizar a sua própria linguagem de programação. Neste trabalho a linguagem utilizada é a PDE (Programação por Diagrama Esquemático).

A PDE é baseada em um método de programação intuitiva e visual (Dexter, 2002). A programação é feita com a montagem do circuito através do simples arrasto, com o *mouse*, dos ícones que estão dispostos em forma de menu na lateral direita da tela, como pode ser observado na Fig. 3.8.

Os ícones são, na realidade, blocos de funções, precisando o programador configurar as características dos mesmos. Uma vez dispostos na área de programação, deve-se realizar a união dos blocos, desenhando-se as linhas com as setas do teclado.

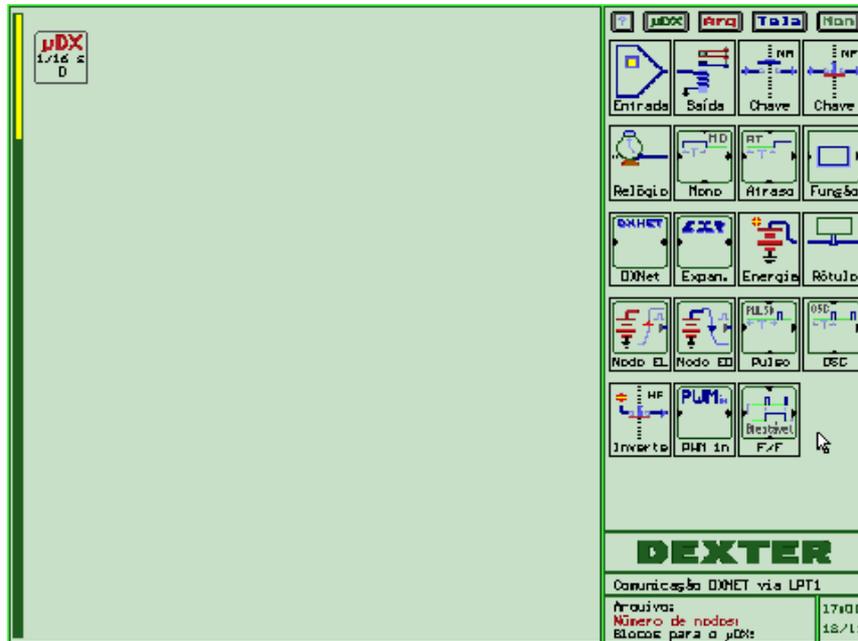


Figura 3.8 - Área de trabalho do Programador Gráfico

Se, por exemplo, quisermos fazer com que um relé temporizado fosse acionado com um determinado tempo de atraso com relação a sua ativação da entrada correspondente, moveríamos para a área de programação uma entrada, um bloco de atraso e uma saída. Após o posicionamento dos mesmos, podem ser desenhadas as linhas de contato, conforme mostrado na Fig. 3.9.

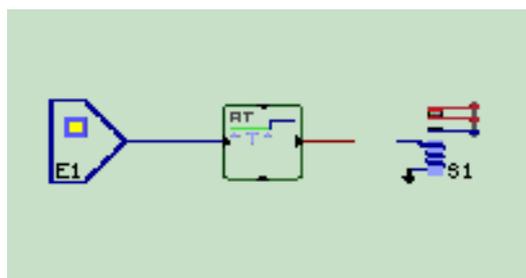


Figura 3.9 - Relé com atraso em relação à ativação da entrada

Para configurar o bloco de funções deve-se manter pressionada a tecla "E" e ciclar com o botão esquerdo do *mouse* sobre o mesmo. No exemplo citado, configura-se o tempo de atraso da função.

Ao implementar-se esta configuração proposta para o controle da célula no CLP algumas modificações foram realizadas. A principal modificação foi decorrente do fato de o programa do CLP trazer consigo entre as funções lógicas uma função capaz de atribuir valores às variáveis do

sistema. Com isso, podem-se distinguir os estados através da atribuição de valores distintos à variável, criando a possibilidade de direcionamento da mesma para condições de troca para o estado que tenha o mesmo valor associado.

Uma das grandes vantagens para a implementação do algoritmo neste CLP é o fato de poderem-se utilizar rótulos que possibilitam a interligação de partes do programa sem a necessidade de desenhar as linhas de conexão. Assim, a existência de dois rótulos com o mesmo valor em pontos distintos do programa equivalem a uma ligação entre os mesmos. Esta capacidade permite deixar a representação gráfica mais enxuta e, por consequência, melhorar sua compreensão. Na linguagem do  $\mu$ DX, os rótulos 0, 1, 2 e 3 são reservados para as ligações com as entradas e os rótulos 4, 5, 6 e 7 para as saídas. Já o rótulo 62 é o terra e o 63 é a energia (Dexter, 2002).

No programa elaborado para o comando criou-se uma memória RS para cada saída. Esta pode ser obtida com uma chave normalmente fechada em paralelo com um curto circuito, o que gera um laço do sinal “set” até que o sinal “reset” seja acionado interrompendo-o. A ação de “reset” desliga o relé de saída, conforme mostra a Fig. 3.10. Optou-se utilizar como sinal “set” os rótulos 8, 9 e 10 e como sinal reset os rótulos 10, 11 e 12 para as saídas S1, S2 e S3, respectivamente.

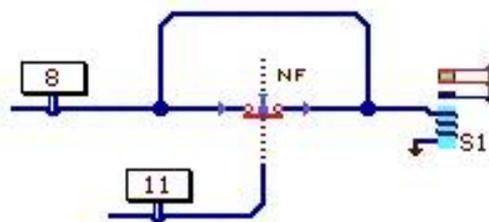


Figura 3.10 - Memória RS ligada à saída S1.

Para iniciar o programa foi utilizado um bloco que produz um pulso quando o CLP é energizado. Este, por sua vez, atribui o valor 0 à variável v0, conforme Fig. 3.11.

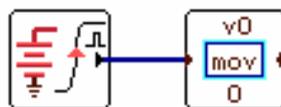


Figura 3.11 - Atribuição inicial a variável v0

Com isso energiza-se a segunda linha que contém as condições de troca para o estado de abertura da porta, que no caso são E1 e E3. Optou-se neste momento por utilizar apenas estes dois sinais como condições de troca de estado, pois nesta configuração a seqüência não só é iniciada com a porta fechada, mas também com a porta aberta ou entreaberta.

Quando satisfeitas estas condições, é atribuído à variável  $v0$  o valor 1 e esta, por sua vez, será direcionada à linha que tem o mesmo valor atribuído à variável, realizando a respectiva ação. Nesta mesma linha estão dispostas as condições para a troca para o próximo estado, como pode ser observado no programa da seqüência apresentado na Fig. 3.12.

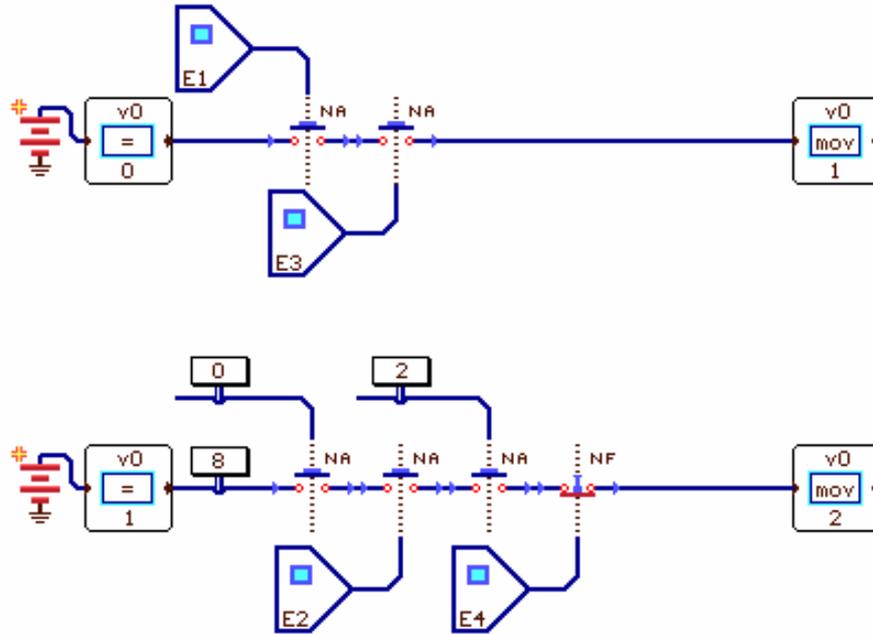


Figura 3.12 - Linha 1 e Linha 2 do programa de controle seqüencial.

As condições de troca, neste caso, foram colocadas na mesma seqüência da Tabela-Verdade anteriormente apresentada. Pode-se observar que a combinação correta permite, através do acionamento das chaves dispostas, o fechamento do circuito que leva a atribuição do novo valor à variável. Para tanto, coloca-se uma chave normalmente aberta para o sinal que deve estar ligado e uma chave normalmente fechada para o que deve estar desligado. Por questão de segurança, colocou-se todo o conjunto de sinais como condição de troca de estado para que antes que cada troca ocorra haja uma varredura que possibilite a localização de possíveis falhas provenientes dos dispositivos ou do meio físico de comunicação.

O programa completo pode ser visualizado no Anexo 3.

#### 4. EXPERIMENTOS DE CAMPO

Uma vez montado o programa, pode-se testá-lo no próprio programador visual do CLP que possui um módulo simulador. Com a depuração do algoritmo até a obtenção dos resultados desejados, partiu-se para um primeiro teste físico. Para tanto, foi construída uma botoeira que

utiliza uma bateria de 9V como fonte de alimentação e possui quatro botões que foram utilizados para simular os sinais de entrada. Com o auxílio da mesma, pôde-se fazer um teste visual do funcionamento do comando lógico devido ao fato de o CLP ser provido de *leds* nas entradas e saídas.

Uma vez comprovadas a eficiência e a robustez do programa, partiu-se para a implementação física do mesmo.

Durante o estudo dos dispositivos, constatou-se nova restrição. Mediante consulta ao manual do CNC (ROMI, 1996) e contato com o fabricante, descobriu-se que o comando presente neste equipamento não possui canal para troca de sinais internos e externos com outros dispositivos. Sendo assim, para fins de desenvolvimento da metodologia, substituiu-se os sinais provenientes do CNC e dos sensores fim de curso pela botoeira. Com esta nova restrição, o sistema a ser testado ficou restrito ao CLP, ao Robô e à botoeira.

Para testar o sistema, conectou-se a saída digital “do8” do robô à entrada E3 do CLP. Escolheu-se esta saída devido ao fato de a mesma estar disponível por ser constantemente utilizada para o acionamento da garra pneumática do robô. Nas demais entradas conectaram-se botões da botoeira para a simulação dos sinais provenientes do CNC e dos sensores fim de curso da sua porta.

Como previamente o robô já tinha acoplado a si uma botoeira, responsável pelo acionamento de quatro sinais de entrada, resolveu-se, substituir o botão responsável pela entrada “di4” pelo relé da saída S2 do CLP.

De posse destes dados, editou-se um programa na linguagem “Rapid”, utilizada nos robôs ABB (ABB,1994). Ao ser iniciado este programa move o robô para um ponto de descanso, ajusta o sinal de saída “do8” e permanece em modo de espera até que o sinal de entrada “di4” seja igual a 1. Neste instante, o sinal de saída “do8” será zerado pelo programa. Com isso o CLP recebe a informação de que o robô está atuando. Após o robô efetuar a ação do programa e retornar ao *setpoint*, “do8” será ajustado novamente pelo programa, caracterizando a troca de estado. Estipulou-se para o programa em questão que seu término ocorreria quando o sinal de entrada “di3” fosse igual a 1, o que pode ser realizado através da botoeira do robô ou pelo próprio controle.

Como ação programada para o robô optou-se pelo desenho de um triângulo sobre uma mesa, uma vez que, com as restrições, não há a necessidade de realizar uma tarefa específica.

Após a realização de todas as conexões, foram acionadas todas as combinações responsáveis pela seqüência com auxílio da botoeira e das reações do robô ao comando do CLP.

## 5. RESULTADOS

Uma vez montadas todas as conexões observou-se um funcionamento sem falha do comando em todas as seqüências realizadas. Também pôde-se observar que o comando não muda de estado com o acionamento de combinações diferentes das que caracterizam a condição para a troca de estado, garantindo assim a segurança do sistema

## 6. CONCLUSÃO

O fato de alcançar-se o funcionamento pleno do comando na primeira tentativa de implementação reforça a importância da dedicação da maior parte do projeto ao planejamento e à edição do comando lógico. Destaca-se também a importância da utilização de vários métodos gráficos distintos, mas complementares, para um melhor entendimento do problema.

Acredita-se que a implementação de um comando lógico simples através da obtenção das funções lógicas é inviável para o controle de sistemas seqüenciais devido à grande dificuldade de edição do controle. Para estes casos considera-se de vital importância para a obtenção dos resultados desejados a utilização das técnicas específicas para comandos seqüenciais como Diagrama de Estados, Diagrama Trajeto-Passo e, principalmente, o Diagrama Grafcet.

Observou-se que o Diagrama de Estados e o Diagrama Trajeto-Passo são de suma importância para a construção da Tabela verdade por fornecerem uma melhor definição e caracterização da seqüência. Já o Diagrama Grafcet torna-se imprescindível para a edição do programa de controle devido ao fato de ele ser uma representação gráfica do mesmo, sendo estes conhecimentos de grande importância prática.

Os resultados obtidos garantem que o comando editado poderia ser implementado na célula em questão se não fossem as restrições impostas pelo próprio CNC. Além disso, com algumas modificações será possível monitorar a célula, junto com outras células comandadas por CLPs, uma vez que o CLP utilizado permite conexão a até outros 14 controladores, via rede DXNET (Dexter, 2002) usando topologia tipo barramento.

A impossibilidade de implementação do controle no sistema completo, devido as restrições do comando do CNC, poderá ser superada através da troca do comando do CNC ou do acréscimo à sua arquitetura atual, de dispositivos que permitam a sua comunicação.

## 7. REFERÊNCIAS BIBLIOGRÁFICAS

ABB Robotics Products, Västerås, Suécia; 1994. **Product Manual IRB 1400.**

Black, J. T., 1991. **The Design of the Factory with a Future.** McGraw Hill, New York.

Bollmann, A., 1996. **Fundamentos Da Automação Industrial Pneumática: Projetos de Comandos Binários Eletropneumáticos.** ABHP, São Paulo.

Dexter, Ind. E Com. De Equipamentos Eletrônicos, Porto Alegre, 2002. **Controlador Programável – Manual de Utilização.**

PROFIBUS, 2002. <http://www.profibus.com>

Richter, C., 2001. **Controladores Programáveis – Curso de Automação Industrial.**

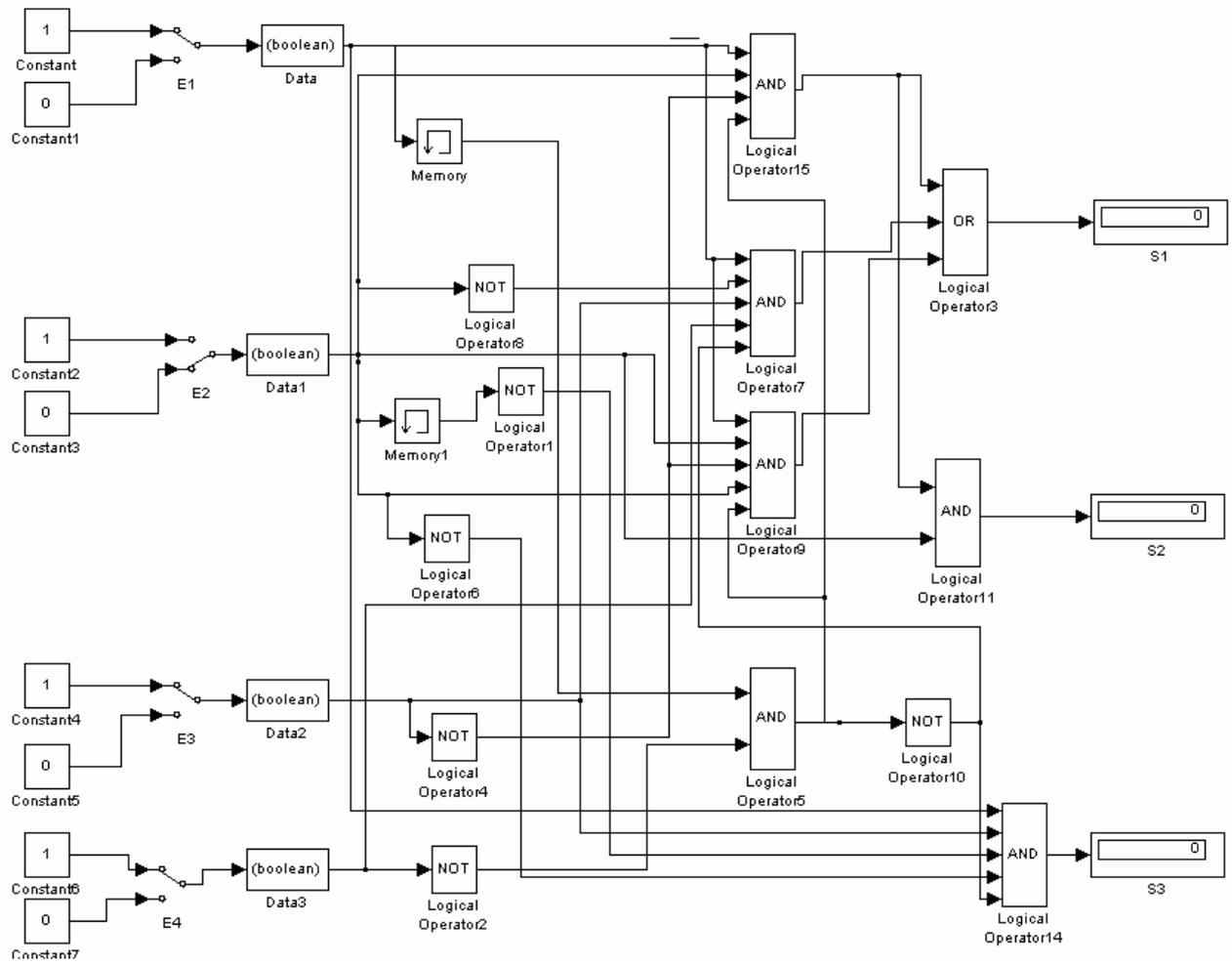
Romano, V. F., 2002. **Robótica Industrial: Aplicação na Indústria de Manufatura e de Processos.** Ed. Edgard Blücher, São Paulo.

ROMI, Industria S/A, São Paulo, 1996. **Manual de Programação e Operação CNC Mach 4.**

Silveira, P. R.; Santos, W. E. dos; 1998. **Automação e Controle Discreto.** Érica, São Paulo.

Simulink, "Dynamic System Simulation Software". User's Guide. Release notes version 1.3. The MathWorks, Inc., Julho 1994.

## ANEXO 1



Comando lógico.

## ANEXO 2

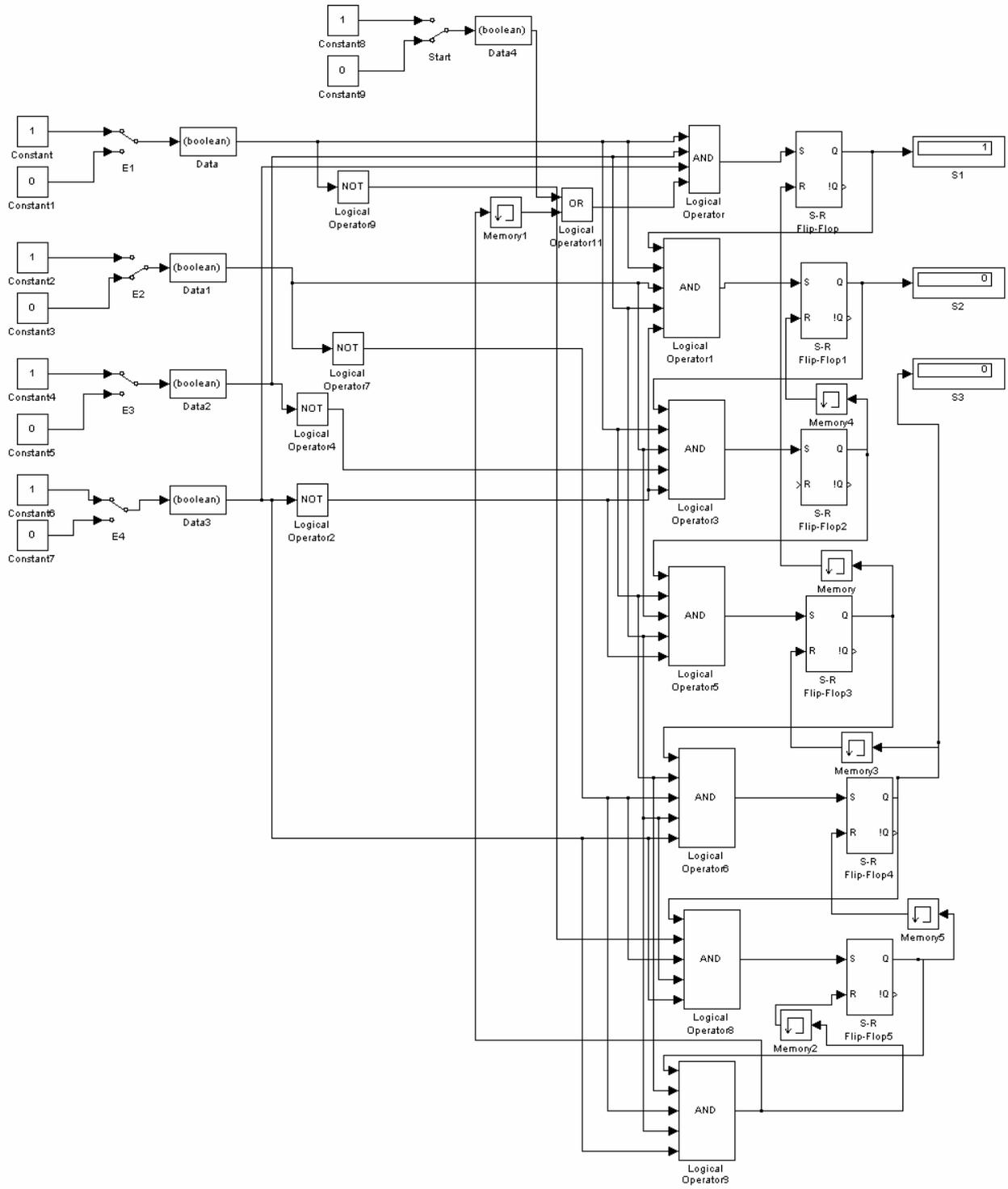
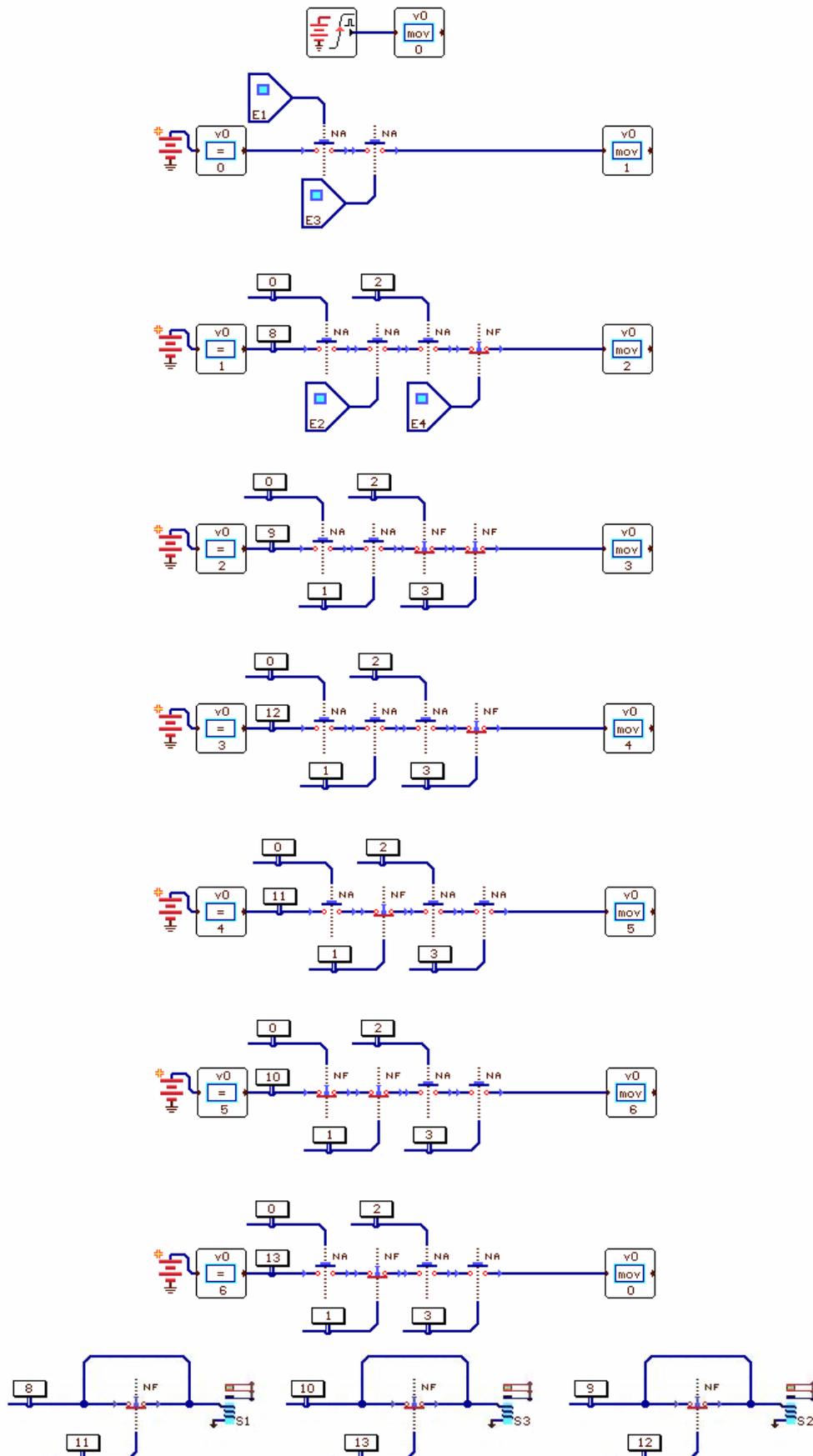


Diagrama lógico seqüencial.

## ANEXO 3



Programa utilizando linguagem do CLP Dexter.