



Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia Mecânica

Interligação Robô – CLP

Projeto de Formatura

Autores:

Eric Preuss

Julio Cesar Roggero

Orientador:

Prof. Dr. Lucas A. Moscato

São Paulo, 14 de dezembro de 1998

8,5 (oitos e cinco)

Dedicatória

*Este trabalho é dedicado ao professor
Lucas A. Moscato , pelas noites perdidas
e intenso bom humor, nas horas mais difíceis*

Índice

1. Motivação.....	6
2. Descrição do Problema.....	8
3. A Alternativa AS-Interface.....	11
4. A Rede PROFIBUS.....	16
4.1 Introdução.....	16
4.2 Ligação/Conexão - Camada 1.....	19
4.2.1 Parâmetros do Fio.....	20
4.2.2 Definição dos Pinos da RS485.....	22
4.3 Protocolo de Acesso de Dados da PROFIBUS-DP - Camada 2.....	23
4.3.1 Estrutura do Telegrama.....	24
4.4 A Inviabilidade do uso da rede PROFIBUS.....	27
5. Robô ASEA.....	29
5.1 Introdução.....	29
5.2 Funcionamento.....	31
5.2.1 Modo de Operação Manual.....	31
5.2.2 Modo de operação automático.....	32
5.2.3 Movimentação.....	33
5.3 Entradas e Saídas Digitais.....	34
5.3.1 Dados Técnicos.....	35
5.4 Programação.....	36
5.4.1 Instruções de Posicionamento.....	36
5.4.2 Instruções de Controle.....	37
5.4.3 Programa do Robô.....	38

6. CLP.....	43
6.1 Introdução.....	43
6.2 Características do CLP a ser utilizado.....	43
6.2.1 Características Técnicas das Entradas e Saídas Digitais.....	45
6.3 Programação.....	46
6.3.1 Meios de Programação.....	47
6.3.2 O Programa.....	49
7. Estudo de funcionamento do Robô.....	51
7.1 Movimentação do Robô.....	51
7.2 Estudo de entradas e saídas digitais.....	55
8. Estudo de funcionamento do CLP.....	59
8.1 Introdução.....	59
8.2 Descrição da programação do CLP e suas saídas digitais.....	59
9. Realização de testes no CLP.....	63
10. Realização de testes no Robô.....	69
11. Programa final de comunicação Robô - CLP.....	74
11.1 Descrição dos programas.....	74
11.2 Programa final do CLP.....	80
11.3 Programa final do Robô.....	82
12. Considerações Finais.....	85
ANEXO A – Listagem do Programa inicial no Robô.....	86
ANEXO B – Listagem em <i>Instruction List</i> do Programa inicial no CLP....	87
ANEXO C – Diagrama de Relés do programa inicial no CLP.....	88
ANEXO D – Diagrama das Placas do Robô.....	89
Bibliografia.....	90

Lista de Figuras

Figura 1.1 – Esquema de uma célula de manufatura.....	6
Figura 2.1 – Composição esquemática da célula de manufatura.....	8
Figura 3.1 – ASi e os diversos equipamentos.....	13
Figura 3.2 – Cabo de energia e dados num bloco aplicativo.....	13
Figura 3.3 – Bloco aplicativo conectado ao cabo ASi.....	14
Figura 3.4 – Cabo chegando a um bloco aplicativo.....	14
Figura 4.1 – Áreas de aplicação da rede PROFIBUS.....	17
Figura 4.2 – Esquema de funcionamento geral do sistema.....	20
Figura 4.3 – Cabo e resistências da rede PROFIBUS-DP.....	23
Figura 4.4 – Formato geral do telegrama da rede.....	26
Figura 5.1 – Esquema do robô com seus possíveis GL.....	29
Figura 5.2 – Caixa de aprendizado do Robô.....	31
Figura 5.3 - Modo de operação automático do robô	32
Figura 5.4 – Entrada e saídas digitais do robô	35
Figura 5.5 – Posicionamento do robô para manipular peças.....	39
Figura 6.1 – Ciclo de processamento do CLP.....	43
Figura 6.2 – Esquema do CLP utilizado.....	44
Figura 6.3 – Meios de programação do CLP.....	48
Figura 6.4 – Diagrama do circuito do exemplo.....	49
Figura 7.1 – Sistema de coordenadas cilíndricas do robô.....	52
Figura 7.2 – Sistema de coordenadas retangular do robô.....	53
Figura 7.3 – Sistema de coordenadas da garra do robô.....	54
Figura 7.4 – Esquema da placa DSQC 122.....	55
Figura 7.5 – Ligação entre as placas DSQC 122 e DSQC 124.....	57
Figura 7.6 – Ligação entre as placas DSQC 122 e DSQC 124.....	58
Figura 8.1 – Diagrama de reles para a programação DDC.....	60
Figura 8.2 – Diagrama para a programação DDB.....	60
Figura 8.3 – Esquema representativo do CLP (E/S).....	61
Figura 8.4 – Esquema das saídas do CLP.....	62
Figura 11.1 – Esquema da situação do teste final.....	75

Lista de Tabelas

Tabela 4.1 – Camadas definidas pelo modelo ISO/OSI.....	16
Tabela 4.2 – Parâmetros do fio de Bus.....	21
Tabela 4.3 – Definição dos pinos da interface RS 485.....	22
Tabela 6.1 – Entradas digitais do CLP.....	45
Tabela 6.2 – Saídas digitais do CLP.....	46
Tabela 7.1 – Entradas digitais e respectivos bornes da DSQC 122.....	56
Tabela 7.2 – Saídas digitais e respectivos bornes da DSQC 122.....	56
Tabela 7.3 – Saídas digitais e respectivos bornes da DSQC 124.....	57

1. Motivação

A automação e a robotização, hoje, já são uma realidade nas indústrias do mundo. Para ser competitivo, são necessários três fatores: Preço, Qualidade e tempo; e uma das maneiras mais confiáveis para consegui-los é justamente através da utilização de máquinas automáticas, como robôs, máquinas CNC, etc..

Dentro destes conceitos de automatização, existe algo que foi idealizado, e hoje já é uma realidade em algumas empresas: uma célula de produção automática.

Esta célula de produção automática, já foi proposta como tema para inúmeras Teses de Mestrado na Escola. A célula automática proposta pode ser representada pela figura abaixo.

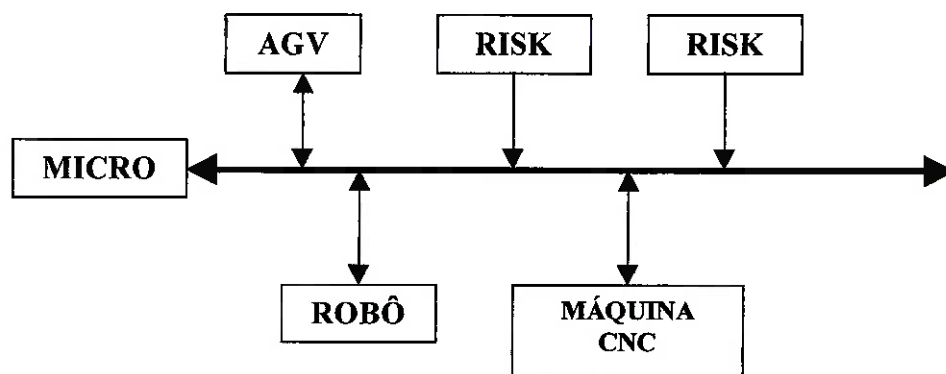


Figura1.1 – Esquema de uma célula de manufatura

Na célula, o funcionamento seria basicamente o seguinte: As estações RISK serviriam para armazenar os softwares de CAD/CAE/CAM responsáveis pelo dimensionamento das peças produzidas e geração de sinais que iriam abastecer a máquina CNC e o Robô. O computador seria o responsável pelo link entre estas estações. O veículo automático (AGV) seria o responsável pela simulação de um transporte de peças dentro da célula ou fábrica. Tal veículo seria controlado por Radio-frequência.

Neste contexto, o Robô seria responsável pelo transporte e posicionamento das peças de uma máquina para outra (no caso de existirem diversas máquinas) e para o box de peças. No caso, o Robô abasteceria a máquina CNC.

Esta célula, chegou a ser implementada, mas nunca chegou a ter um funcionamento 100%, e serviu para base de muitas Teses de Mestrado. O trabalho de formatura teve como motivação inicial esta célula, partindo-se para apenas uma parte simplificada dela.

O tema escolhido ainda teve como motivação uma tentativa de encontrar algo próximo aos estágios realizados pelos alunos, para que no futuro possa ser de grande utilidade. Os estágios relacionam o CLP e as Redes de Dados (ao estágio realizado por um dos alunos na SIEMENS) e a parte do Robô (ao estágio realizado por um dos alunos na General Motors).

2. DESCRIÇÃO DO PROBLEMA

O problema proposto, está diretamente relacionado com a célula de manufatura descrita acima. Na verdade, imaginou-se uma parte da célula acima, composta pelos elementos abaixo:

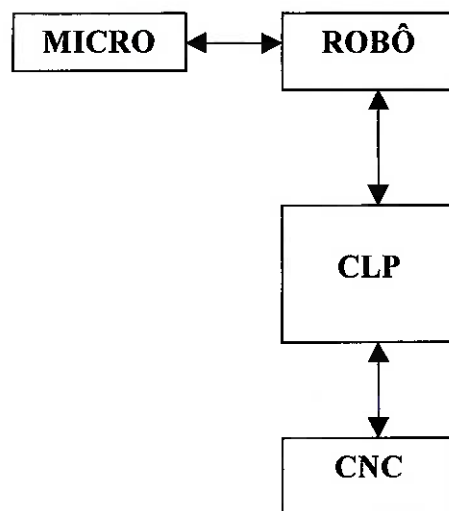


Figura 2.1 - Composição esquemática da célula de manufatura.

O funcionamento da célula seria o seguinte: O micro instalado ao lado do Robô, seria encarregado da programação “off-line” do mesmo (conversão de protocolos). Uma vez feito o programa no Robô, este se comunicaria com o CLP, através de entradas e saídas digitais, enviando e recebendo sinais de sincronismo. O CLP por sua vez, abrigaria não só um programa para controle do Robô, mas também do CNC; Caberá a ele emitir sinais de sincronismo para as operações do Robô e CNC, simulando um ambiente com mais máquinas e operações (ambiente industrial), onde as máquinas devam executar ações e

retornar as ações realizadas, aguardando nova ordem para no momento certo realizar novas operações.

Neste trabalho, será analisada a comunicação do Robô com o CLP. A parte de programação “off-line” do Robô será feita por outro grupo, e a parte de comunicação com o CNC ficaria para um próximo trabalho, já que o CNC não possui (no momento) uma placa de entradas e saídas digitais. O grupo se propõe a resolver o seguinte problema:

- Elaboração de um programa no Robô, que simule uma rotina “pick and place”. O Robô neste caso seria programado no modo manual, conforme será descrito no item apropriado;
- O programa deve utilizar valores que seriam armazenados em algum lugar na memória do Robô, sinais estes advindos da programação “off-line” feita pelo outro grupo, para verificar a validade desta;
- Programação do CLP, a fim de sincronizar os movimentos do robô, de acordo com um sinal enviado através de uma botoeira acoplada ao controlador;
- Interligação do CLP com o Robô e realização de testes que verifiquem o funcionamento do mesmo, evidenciando a comunicação entre os dois equipamentos.

Para realizar esta tarefa de comunicação do Robô e CLP, foram pensadas em algumas soluções possíveis antes da definitiva adotada neste trabalho. Algumas destas soluções pensadas envolviam comunicação

utilizando uma rede AS-Interface (fabricada e utilizada pela SIEMENS na automação fabril) e até mesmo um BUS de dados (rede PROFIBUS, também SIEMENS). Todas essas alternativas foram analisadas e avaliadas na fase de Viabilidade e serão comentadas no decorrer deste trabalho.

3. A Alternativa AS-Interface

A alternativa de se utilizar uma rede ASi (*Actuator-Sensor-Interface*) foi levantada. Pensou-se em utilizá-la pois esta se baseia em um volume de transmissão de dados pequeno, com taxas de transmissão não muito altas. E inicialmente, era exatamente disso que era necessário para a determinação do projeto.

Segue a seguir uma sequência de dados técnicos referentes a esse tipo de rede de dados, a que tipo de aplicação ela se destina e como se dá a sua comunicação com as redes de dados de mais alto nível. Mostra-se também, o porque dessa alternativa ter sido considerada imprópria para a execução deste projeto.

Algumas características da rede ASi podem ser observadas abaixo:

- 1- Trabalha com o princípio Mestre-Escravo;
- 2- Pode-se utilizar mais de 31 Escravos;
- 3- Cada escravo pode ter 4 entradas digitais e 4 saídas digitais em 24V;
- 4- Pode-se adicionar 4 bits-parâmetro por escravo;
- 5- Número máximo de 248 entradas e saídas digitais;
- 6- Também é possível utilizar E/S analógicas ;
- 7- Endereçamento automático através da conexão do bus;
- 8- Dois fios sem proteção por cabo;
- 9- Dados e Alimentação em um só cabo;

- 10-Comprimento máximo de 100m por linha podendo estender a 300m com uso de repeaters;
- 11-Não é necessária a utilização de resistores no final do bus;
- 12-Tempo de ciclo < 5ms.

Normalmente, as informações e a energia são transmitidos em um mesmo cabo (cabo amarelo). Em caso de emergências ou de necessidade de mais energia, pode-se utilizar uma fonte adicional de energia com cabos adicionais (preto - DC ou vermelho - AC).

A rede ASi pode ser programada por um PLC, que possua uma placa apropriada para sua utilização. Esta rede pode ser ligada a uma rede de mais alto nível conhecida com PROFIBUS - DP, que será descrita posteriormente.

De maneira geral a rede ASi é destinada a aplicação com sensores e chaves binárias, existindo botoeiras destinadas à mesma, bem como chaves fim de curso, conforme mostra a figura 3.1.

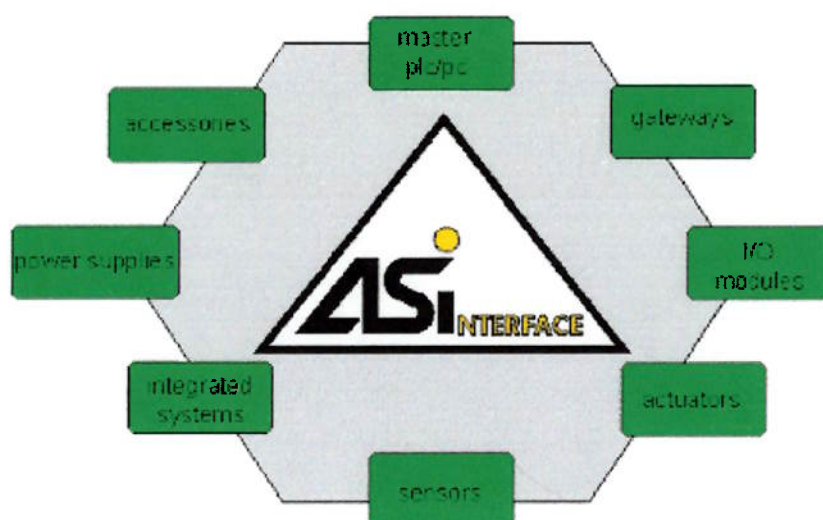


Figura 3.1 - ASi e os diversos equipamentos que podem fazer parte do conjunto.

O cabo deste tipo de rede é mostrado na figura 3.2, juntamente com a sua conexão num bloco aplicativo.

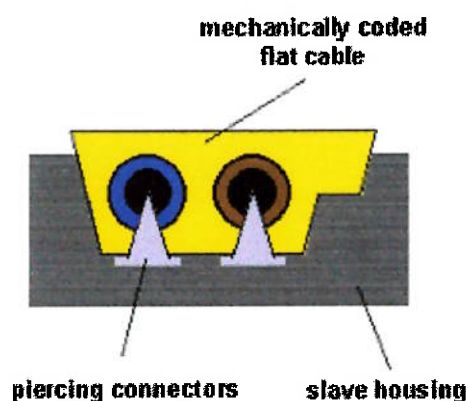


Figura 3.2 - Cabo de Energia e Dados num bloco aplicativo

A figura 3.3 mostra uma conexão real realizada com o cabo e um bloco aplicativo de entradas e saídas digitais, utilizando o cabo apropriado.

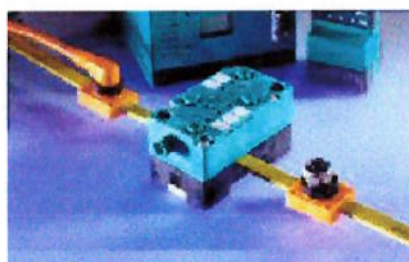


Figura 3.3 - Bloco aplicativo conectado ao cabo ASi.

A figura 3.4 já mostra as 4 E/S digitais esquematizadas para recepção com um sensor.

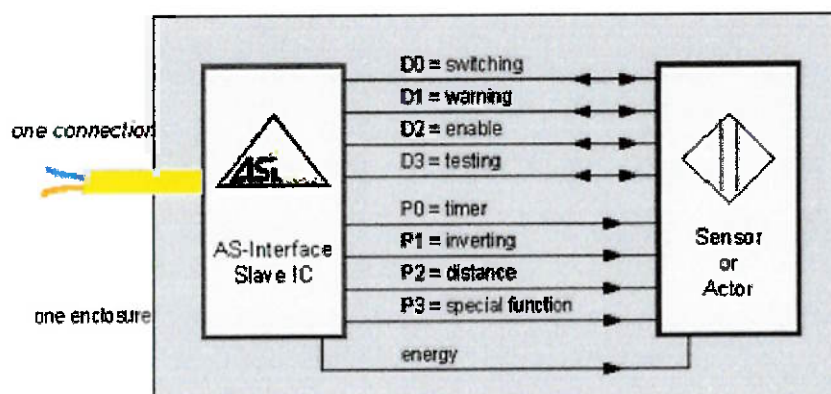


Figura 3.4 - Cabo chegando a um bloco aplicativo e deste distribuem-se as E/S digitais.

Inicialmente pode-se notar que esse tipo de rede pode muito bem se adequar às necessidades deste projeto, porém caiu-se em um problema: este

tipo de rede não se refere a uma comunicação para um Robô, o qual é o enfoque deste trabalho. E é a partir deste princípio que pode-se observar outros problemas como a taxa de transmissão de dados e a dificuldade em se obter a informação precisa. Um problema deste tipo de rede é não ter a certeza da chegada da informação partindo do PLC, já que não se tem uma segurança de dados, como é enunciado no item 8. Torna-se difícil então saber se o que foi transmitido é um dado seguro. A segurança no caso de uma operação com um Robô é de extrema importância, já que dependendo do que se vai transmitir, o Robô pode vir a executar movimentos perigosos, tanto ao equipamento como a supostos operadores que estejam por perto.

Apesar do programa do Robô ser armazenado diretamente na memória do mesmo, e o PLC, inicialmente, somente mandar sinais de sincronismo ao Robô não existindo grande perigo neste caso (situação controlada), há um erro conceitual em utilizar uma rede ASi, para uma aplicação que não seja de atuadores e sensores binários como é enunciado.

4. A Rede PROFIBUS

4.1 Introdução

Quando se pensa em sistemas de comunicação e compara-se com outros nota-se que é extremamente importante padronizar um modelo de referência. O modelo ISO/OSI (*International Standards Organization/Open Systems Interconnect*) foi criado em 1978 como um modelo internacional de referência. Esse modelo divide o processo de comunicação em 7 camadas conforme mostrada abaixo:

Tabela 4.1 - Camadas definidas internacionalmente pelo modelo ISO/OSI.

Camada 7	Camada de Aplicação
Camada 6	Camada de Representação
Camada 5	Camada de Sessão
Camada 4	Camada de Transporte
Camada 3	Camada de Rede
Camada 2	Camada de Conexão de Dados
Camada 1	Camada Física

A Rede PROFIBUS está presente nas camadas indicadas em negrito. Mas é claro que um sistema de comunicação industrial deve cobrir uma larga faixa em termos de aplicações industriais, e é baseado nisso que este tipo de rede é dividido em três variações, são elas:

- PROFIBUS - DP: destinada a alta velocidade de comunicação e transmissão de dados, que é requerida na automação de fábricas e automação predial. Por DP entende-se *Decentralized Periphery*.
- PROFIBUS - FMS: por ser orientada ao objeto, tem como principal proposta ser destinada a comunicação de dados em células e em nível de chão de fábrica. Por FMS entende-se *Fieldbus Message Specification*.
- PROFIBUS - PA: reúne os requisitos de um processo industrial e oferece aplicações para áreas de segurança intrínseca e de não segurança intrínseca, como controlar e emitir sinais para o dispositivo de chão de fábrica e permitir a transmissão desses dados para um nível hierárquico mais alto da rede. Por PA entende-se *Process Automation*.

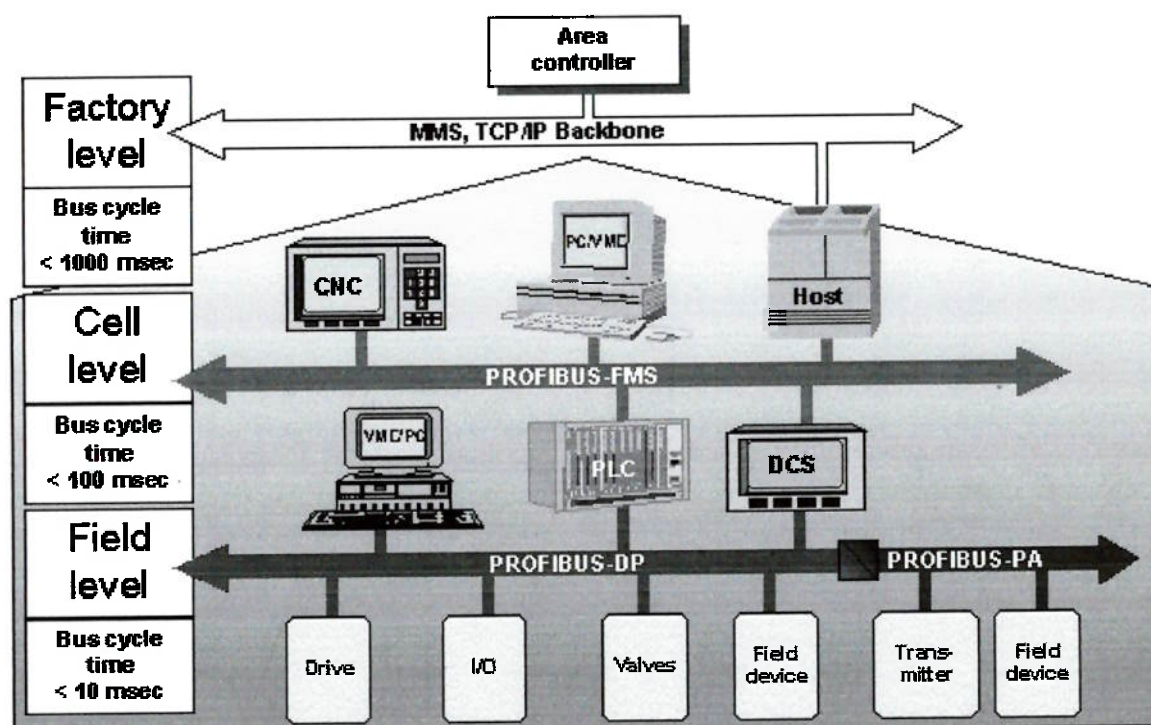


Figura 4.1 - Áreas de Aplicação com indicação dos diferentes ciclos de tempo.

Nota-se, portanto, que para a aplicação em questão a variação recomendada é a PROFIBUS - DP. Desse modo, de agora em diante quando houver referência a PROFIBUS, automaticamente se refere a variação PROFIBUS - DP.

4.2 Ligação/Conexão - Camada 1

A conexão da rede PROFIBUS ao CLP é feita através de uma placa opcional que é acoplada ao CLP de forma a transformar os sinais gerados pelo controlador no formato do padrão PROFIBUS.

A conexão da rede com a interface (que fará a comunicação com o Robô) será efetuada através de um terminal que “transforma” os sinais advindos do cabo da rede para uma saída do tipo RS485. Esta saída RS485 será utilizada para a conexão com a interface a ser definida posteriormente.

A interface montada deverá ser composta imprescindivelmente de um chip que efetua a tradução do protocolo de dados vindo do CLP/Placa no Padrão PROFIBUS. Este chip é especificado de acordo com a taxa de transmissão dos dados que se pretende impor ao sistema e é chamado ASIC (*Application Specific Integrated Circuit*). Pretende-se desta forma executar uma transmissão de dados que possa transmitir ao Robô os sinais de sincronismo, necessários à sua movimentação. Em linhas gerais, pretende-se filtrar o protocolo de dados da forma PROFIBUS, a fim de obter apenas os dados de execução sincronizada. Além disso, o robô efetuará a resposta ao PLC da operação já efetuada e este enviará um novo sinal de execução da próxima tarefa. Portanto, a interface a ser desenvolvida deverá prover sinais filtrados às entradas e saídas digitais do Robô e retornar deste, montando o protocolo completo, o dado de operação efetuada.

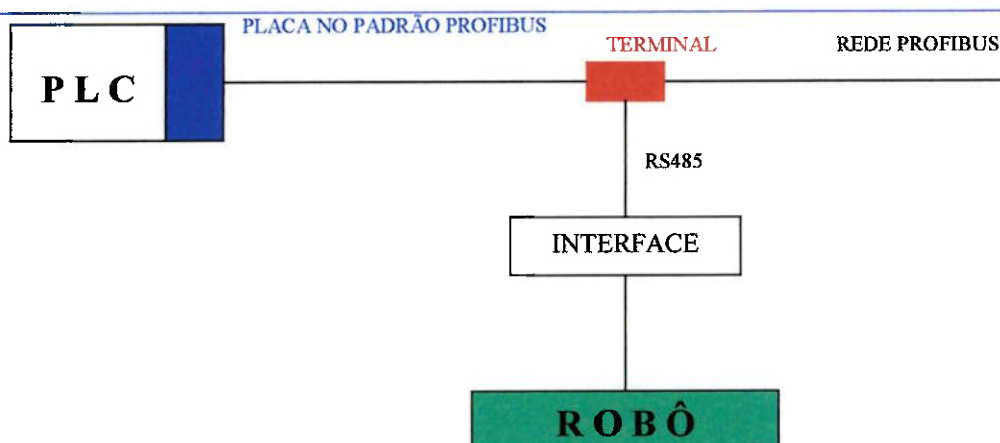


Figura 4.2 - Esquema proposto de funcionamento geral do sistema com a rede.

Para efetuar a montagem dos equipamentos é necessária uma definição completa do ponto de vista físico de montagem da rede, ou seja, a definição da camada 1, já citada.

A montagem física da rede implica na definição do tipo de fio que será utilizado, bem como a definição dos pinos da RS485, além das especificações da norma para com os dados técnicos de impedância, resistência, etc..

4.2.1 Parâmetros do Fio

De acordo com o especificado pela norma (EN 50170), o fio de bus deve ser do tipo A e pode ser usado conforme mostra a tabela 2. Um linha de bus do tipo B também é fornecida porém não tem importância para a aplicação em questão.

Tabela 4.2 - Parâmetros do Fio de Bus.

Parâmetro	Linha A
Impedância (Ω)	135 a 165
Capacitância por unidade de comprimento	< 30 pF/m
Resistência de loop (Ω/km)	110
Diâmetro do fio (mm)	0.64
Área de condução do fio (mm^2)	> 0.34

A especificação dos parâmetros da linha A resultam nos seguintes comprimentos do segmento de bus:

Taxa de Transmissão (kbit/s)	9.60	19.20	93.75	187.50	500	1500	12000
Comprimento (m)	1200	1200	1200	1000	400	200	100

Sendo utilizada a linha A e para taxas de transmissão altas (por exemplo 12MBaud) devem-se usar resistências no final do fio de bus, para que a ação de harmônicos na rede sejam eliminadas. No caso da linha A utilizam-se resistências de 390Ω entre o GND e o fio.

4.2.2 Definição dos Pinos da RS485

Os pinos da interface RS485 são designados conforme a tabela abaixo:

Tabela 4.3 - Definição dos pinos da interface RS485.

Pino	Sinal	Designação
1	Proteção	Terra de Proteção
2	M24	Terra de 24V de tensão de saída
3	RxD/TxD-P	<i>Receive data/transmission data plus</i>
4	CNTR-P	Sinal de controle para repetidores (controle de direção)
5	DGND	Potencial de transmissão de dados (do terra para 5V)
6	VP	Tensão de alimentação da resistência final (P5V)
7	P24	Tensão de Saída +24V
8	RxD/TxD-N	<i>Receive data/transmission data negative</i>
9	CNTR-N	Sinal de controle para repetidores (controle de direção)

VP deve ser capaz de controlar uma corrente mínima de 10mA para manobrar os resistores finais.

Linhas curtas não devem ser usadas com alta taxa de transmissão. A tolerância para cargas capacitivas deve estar entre 15 e 25pF.

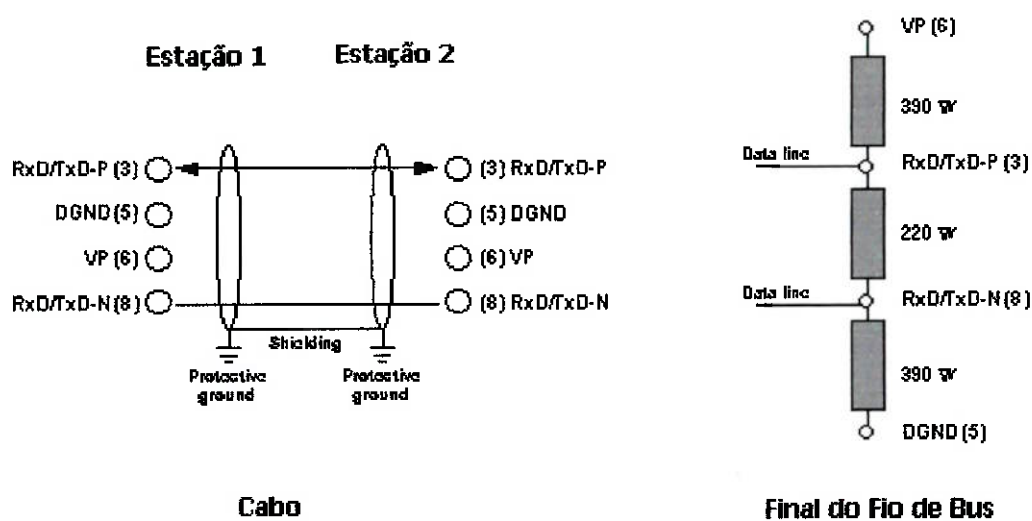


Figura 4.3 - Cabo e Resistências de fim de linha da rede PROFIBUS - DP.

Existe a possibilidade de transmissão por fibra óptica, porém a aplicação não necessita deste tipo de transmissão.

4.3 Protocolo de Acesso de Dados da PROFIBUS - DP - Camada 2

Estes tópicos de definição do Protocolo de Acesso são meramente informativos, já que os chips ASICs por serem dedicados já fazem todas as etapas do protocolo, porém o conhecimento dessas etapas é de suma importância para localização e solução de possíveis problemas advindos da implementação da interface e da execução da troca de informações entre as duas estações em questão (PLC e Robô).

A estrutura do protocolo de dados da rede PROFIBUS é definida dentro da camada 2 (Conexão de Dados). O padrão ISO/OSI define o formato geral dos telegramas usados na transmissão de dados, os mecanismos de

segurança e os serviços de transmissão, entre outros. Os serviços de transmissão DP são definidos conforme abaixo:

- **SRD:** *Send and Request Data with acknowledge*. Dados são enviados e recebidos em um ciclo de telegrama.
- **SDN:** *Send Data with No acknowledge*. Transmissão ou multitransmissão de telegramas, ou seja, telegramas para um grupo selecionado.

Durante o serviço SRD, o mestre envia dados de saída para o escravo e recebe dados de entrada em resposta durante um período especificado de tempo. Se o escravo é somente um dispositivo de saída, este responde com um *short acknowledgment* "E5H". O serviço SDN envia dados para um grupo definido de equipamentos escravos. Existe *no acknowledgement* para um telegrama SDN.

4.3.1 Estrutura do Telegrama

A estação inativa envia inicialmente um tempo de sincronismo do envio de dados (33 Tbits), que deve ser definido antes de iniciar-se a transmissão dos dados propriamente dita. Todos os dados são transferidos sem lacunas entre os caracteres individuais. Os ASICs já executam esta regra.

O significado das abreviações dos telegramas são listados abaixo:

- **SD:** *Start Delimiter* (para distinção entre os diferentes tipos de telegramas)

- FC: *Function Code* (para identificar o tipo de telegrama de acordo com o requisitado, telegramas de entendimento ou de resposta)
- LE: comprimento dos dados de rede: DA, SA, FC, DSAP, SSAP
- LEr: Repetição do comprimento após a não inclusão em HD = 4 (Hamming distance)
- DA: endereço de destino
- SA: endereço da estação de envio
- DSAP: *Destination Service Access Point*. A estação destino usa o DSAP para determinar que serviço deverá ser executado
- SSAP: *Source Service Access Point*
- DU: *Data Unit* (dados da rede, valores entre 1 e 244 bytes)
- FCS: *Frame Checking Sequence* (adição de bytes com o tamanho especificado. Poucos ASICs podem fazer isso)
- ED: *End Delimiter* (sempre 16H)

A camada de transmissão de dados também inclui transmissão segura de dados. O ED é especificado como 10H e não precisa ser incluído no telegrama novamente. A *Hamming distance* (HD = 4) define uma divisão em quatro regiões bem determinadas do telegrama para que se possa detectar erros na transmissão e permitir uma ação corretiva.

Os erros que podem ser detectados com HD = 4, são:

- Erro no formato do carácter (paridade, sobreposição e erro nas bordas do protocolo);

4.4 A Inviabilidade do uso da rede PROFIBUS

Vários são os motivos para que a continuidade deste projeto, com vistas a utilização da Rede PROFIBUS, venha a se tornar inviável, porém o mais forte deles é que tem-se a necessidade de receber apenas alguns bits de dados para a movimentação (sincronismo) do Robô. Como mostrado no item anterior a rede PROFIBUS necessita em seu protocolo uma transmissão de cerca de 256 bytes/ciclo para efetuar o link de dados e manter comunicação entre as duas estações (PLC e Robô). Nota-se que isso torna-se totalmente inviável, a medida que se pretende efetuar a comunicação entre as duas estações de forma a mostrar que o Robô recebeu uma informação, processou e enviou outra ao PLC possibilitando a este emitir nova ordem de trabalho. Além disso, apesar de tratar-se apenas de um protótipo de utilização da rede ligada a um PLC e um Robô, o custo para adquirir um PLC com uma placa PROFIBUS mais o fio de bus, mais o chip para construção e projeto da interface, além dos outros componentes necessários à montagem do circuito, bem como a placa de circuito impresso a ser desenvolvida, fez com que o projeto com a aplicação da rede PROFIBUS se tornasse totalmente desnecessário para este nível de aplicação.

Outros detalhes da Rede Profibus foram estudados e pesquisados, como os tipos de chip ASIC e suas aplicações (diversas taxas de transmissão, por exemplo), Controladores, Endereçamentos, estruturas dos buffers, cálculo do tamanho da mensagem (telegrama) e ciclos de bus (tempo de resposta de

acordo com o número de DP - Slaves pendurados na rede), a sequência de transmissão dos telegramas e sua estrutura completa.

Além disso, pesquisou-se a respeito de *Profiles* da PROFIBUS - DP, que descrevem como diferentes tipos de robôs podem ser controlados com a PROFIBUS - DP, porém não se chegou a estudá-las, já que se chegou a inviabilidade da continuidade do projeto proposto.

- Erro de protocolo;
- Delimitações de início e fim erradas;
- Erro na checagem do byte estrutural;
- Erro no comprimento do telegrama.

A figura abaixo mostra a necessidade de dados para cada transmissão.

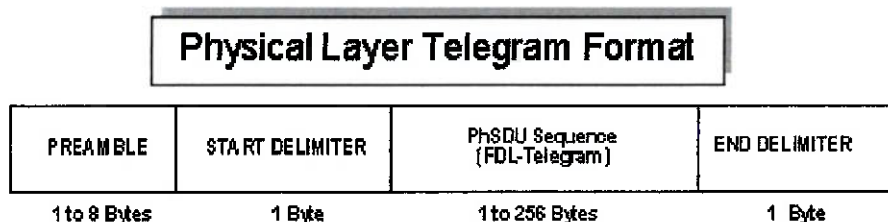


Figura 4.4 - Formato geral do Telegrama da Rede.

Nota-se que há inúmeras possibilidades para o tamanho geral do Telegrama, já que existem quatro regiões bem definidas do mesmo e estas podem variar em seu tamanho, de acordo com a instrução que será realizada ou o tipo de transmissão a ser executado.

É baseado nesta estrutura do telegrama que chegamos a inviabilidade de continuidade deste projeto utilizando este tipo de rede.

5. O Robô ASEA

5.1 Introdução

O Robô ASEA (IRB-6) será utilizado neste trabalho com a finalidade de se mostrar a troca de informações dentro de uma célula de manufatura, fato que é fundamental na automação hoje em dia.

O Robô IRB-6 é um robô com 6 graus de movimento, sendo que cada um é comandado por um dispositivo independente, conforme mostra a figura abaixo:

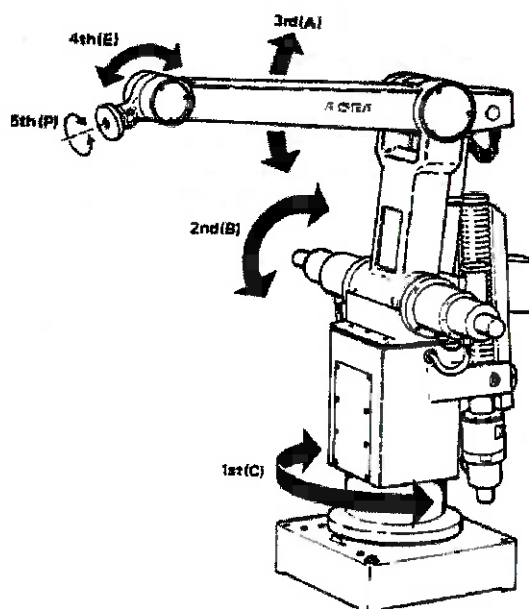


Figura 5.1 - Esquema do Robô com seus possíveis graus de liberdade.

Existem disponíveis diversas formas de se comunicar o robô com outros equipamentos. Uma delas, consiste em ligá-lo a outro computador (denominado *Superior Computer*) que fará a programação Off-line do robô. Para tanto, são necessários dois protocolos: o de comunicação (ADLP10) e o de Aplicação (ARAP) que devem ser implementados em ambos. Esta forma de ligação é base de estudo de outro grupo e portanto não será abordada neste trabalho, mas sim uma outra mais comumente utilizada na indústria .

Nesta outra forma de comunicação com o robô, ele não é programado externamente, sendo que o programa (a ser executado pelo robô) é feito diretamente na unidade de programação do robô, e ele se comunica com equipamentos externos, no caso o CLP, através de entradas e saídas digitais.

Essas entradas e saídas digitais são utilizadas para receber e enviar sinais de sincronismo, simulando-se um ambiente industrial mais amplo, onde existem vários equipamentos e várias tarefas, que estariam relacionadas através de uma dada seqüência de operações. O CLP iria se situar em um nível superior e comandar a sincronização destes equipamentos.

5.2 Funcionamento

O funcionamento do robô ASEA pode-se dar em duas formas:

- Manualmente (usando o joystick presente no painel de controle);
- Automaticamente (usando um programa que está na memória).

5.2.1 Modo de Operação Manual

O modo de operação manual consiste em movimentar o robô através do joystick presente na caixa de aprendizado que está na cabine de controle do robô.

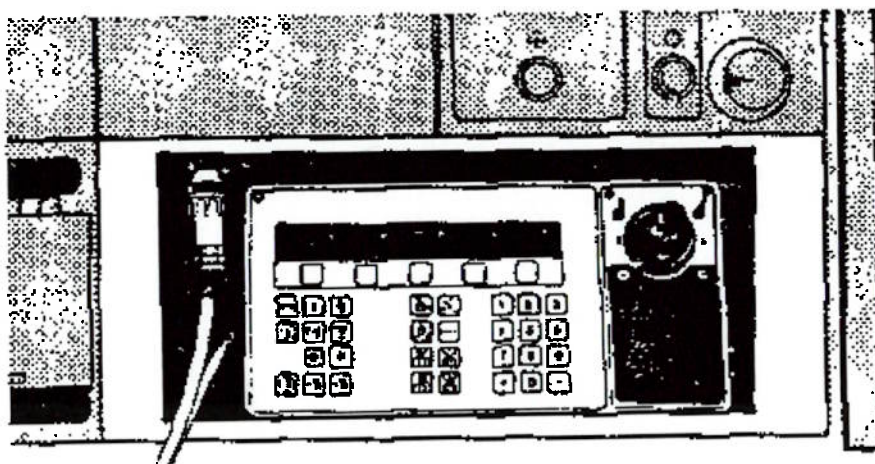


Figura 5.2 - Caixa de aprendizado utilizada para comandar o Robô.

Este modo de operação será usado como uma forma auxiliar, para operação em modo automático, ou seja, o robô será deslocado até a posição desejada utilizando-se este tipo de movimentação e esta posição será armazenada no programa.

5.2.2 Modo de operação automático

O modo de operação automático consiste em se executar um programa que esteja na memória do robô. Antes de se executar este programa, este deve ser carregado na memória e para tanto existem três formas:

- Carregá-lo de um disquete para a memória;
- Programação Off-line;
- Entrar o programa para a memória, uma instrução por vez usando a unidade de programação. Este é o método utilizado para se criar um programa.

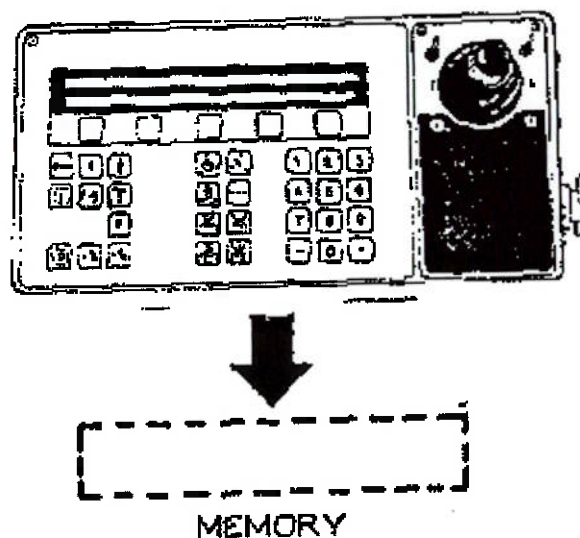


Figura 5.3 - Modo de operação automático com armazenamento de programa na memória

Uma vez feito o programa, há dois modos de se executá-lo:

- *Execução contínua:* na execução contínua o sistema executa o programa inteiro do início ao fim, ininterruptamente e de forma cíclica até que este receba ordem de parada (botão de parada da cabine);
- *Execução Passo a Passo:* neste modo de execução executa-se uma instrução do programa por vez; o programa muda de instrução cada vez que a tecla do painel de controle é apertada.

Este será o modo utilizado a fim de se testar a comunicação entre o CLP e o Robô ASEA.

5.2.3 Movimentação

A movimentação do robô pode-se dar através de 5 possíveis eixos de movimento. Além destes 5 eixos de movimentos, existe ainda mais um grau de movimentação, no dispositivo fixador da garra do robô (uma espécie de punho). Com estes 6 graus de movimentação é possível se levar a robô para qualquer posição no espaço dentro de seu raio de alcance.

Para se mover o braço do robô para um ponto particular no espaço, deve-se especificar a precisa localização daquele ponto, através da movimentação de um eixo de movimento por vez. A movimentação do robô é feita ponto a ponto, e para cada ponto uma instrução diferente, no programa, é requerida.

Através de diversas dessas instruções será feita uma rotina que simule uma operação “Pick-and-place”, que consiste basicamente em se pegar um objeto em determinado lugar e posicioná-lo em outro. Esse tipo de movimentação é de comum uso nas indústrias hoje em dia, e a idéia do trabalho é que no futuro, seja feita o posicionamento de uma peça dentro de uma máquina de operação automática, como por exemplo uma CNC.

5.3 Entradas e Saídas Digitais

O sistema do robô e CLP irão se comunicar através de entradas e saídas digitais programáveis. O sistema básico do robô contém:

- 7 entradas digitais através das quais eventos externos podem afetar operações programadas no robô;
- 6 saídas digitais através das quais o sistema do robô pode controlar equipamentos periféricos;
- 2 saídas digitais para controle da garra.

No caso do estudo em questão, serão utilizados apenas algumas dessas entradas e saídas digitais, para envio de sinais de sincronismo entre o CLP e o Robô.

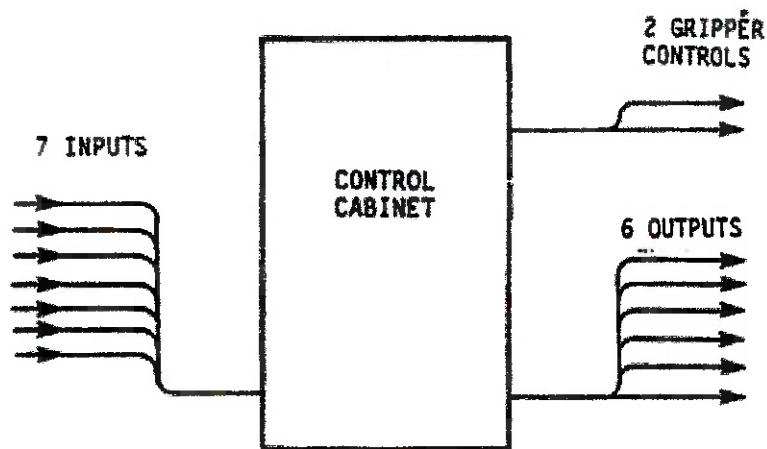


Figura 5.4 – Entradas e saídas digitais do Robô

5.3.1 Dados Técnicos

As entradas e saídas digitais estão numa placa conectada ao controle do robô. Os dados técnicos dessas entradas e saídas são listadas abaixo:

- Tensão Nominal: 24V DC
- Faixa do sinal de entrada:
 - Lógica "1": de 18 a 35V
 - Lógica "0": de -21 a +2V
- Corrente de entrada na tensão nominal: 6.8mA
- Impedância de Entrada: 3.5k Ω
- Sinais de saída:
 - Fonte externa: 24V nominal (20 – 35V)
 - Capacidade de carga: no máximo 150mA

5.4 Programação

A programação do Robô será feita utilizando-se a unidade de programação (vide item 5.2.2). O programa a ser feito no Robô visa além de testar a comunicação com o CLP, também na medida do possível testar a parte de programação Off-line realizada pelo, outro grupo.

Para a simulação de atividades realizadas em uma célula de manufatura (no caso representada por uma rotina "Pick-and-place") serão utilizados basicamente dois tipos de instruções: instruções de posicionamento e instruções de controle.

5.4.1 Instruções de Posicionamento

Uma instrução de posicionamento consiste em se determinar um ponto de posicionamento do braço do Robô (através do uso do joystick), para o qual este deve ser movido durante a execução do programa. Além do ponto de destino, deve ser especificada uma velocidade para cada instrução de posicionamento. Na verdade, para ser tornar o procedimento mais fácil, é especificada uma velocidade básica e as instruções de posicionamento recebem uma porcentagem dessa velocidade como argumento.

Além da velocidade, deve-se especificar um certo grau de acuracidade com o qual se deseja atingir o ponto. Neste sentido, existem dois graus possíveis: acuracidade fina e grossa.

5.4.2 Instruções de Controle

As instruções de controle, neste caso, serão utilizadas com a finalidade de atuar sobre as entradas e saídas digitais. Neste sentido, são oferecidas as instruções: *WAIT*, *JUMP* e *INTERRUPT*.

- *WAIT*: pode ser utilizada para atrasar a execução da próxima instrução do programa até que ocorra um particular evento. O evento pode ser:
 - Espera de um tempo específico;
 - Estado de uma entrada digital.
- *JUMP*: esta instrução é utilizada para controlar a sequência de execução das instruções em um programa. Desse modo, existem dois argumentos para esta instrução:
 - *JUMP* incondicional;
 - *JUMP* condicional. Neste caso, a realização do “salto” para a posição indicada do programa, pode estar condicionada ao estado de uma entrada ou saída digital, ou ainda ao valor armazenado em um registrador.
- *INTERRUPT*: esta instrução é usada para o controle direto de entradas disponíveis como opcionais. Existem três tipos de interrupção:
 - Interrupção imediata da instrução, seguida da execução de uma próxima instrução;
 - Parada do programa quando a instrução corrente é executada;

- Chamada de um subprograma quando a instrução corrente é executada.

5.4.3 Programa do Robô

A idéia inicial do programa do robô, era de se fazer um programa que executasse a sequência abaixo; Essa idéia foi proposta e seu programa idealizado na 1ª parte do trabalho de formatura. Com o passar do tempo, a idéia foi aprimorada e novos programas surgindo. O passo inicial foi o programa abaixo:

- Move o robô para um determinado ponto sobre a posição M1;
- Avisar CLP da execução do movimento;
- Receber sinal do CLP;
- Executar movimento para pegar a peça em M1, conforme sinal recebido;
- Move o robô para um determinado ponto sobre a posição M2 (checagem do sinal recebido pelo outro grupo na programação Off-line);
- Avisar CLP da execução do movimento;
- Receber sinal do CLP;
- Executar movimento para soltar a peça em M2, conforme sinal recebido;
- Retorna para o início do programa.

Com as instruções acima, espera-se executar movimentos semelhantes ao mostrado na figura abaixo, com a finalidade de se testar o sincronismo das informações entre o CLP e o Robô.

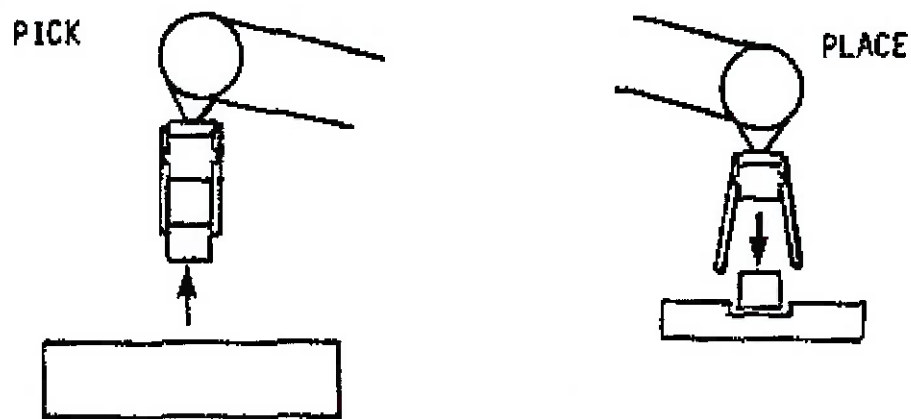
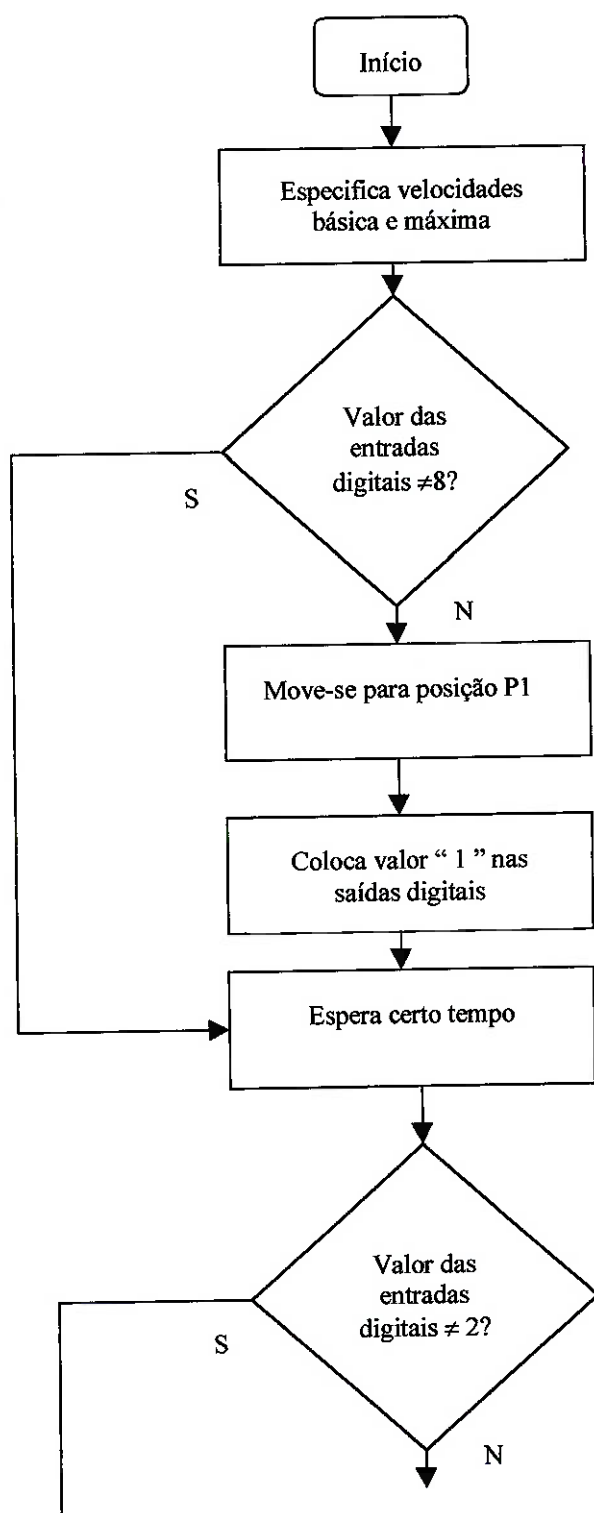
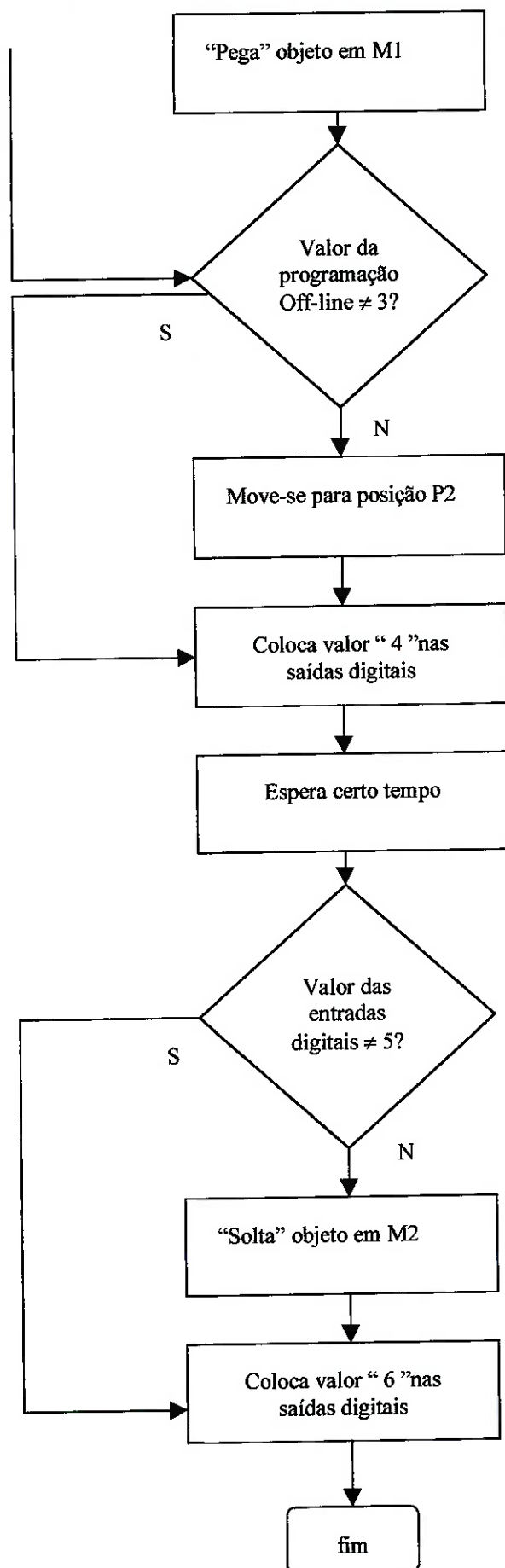


Figura 5.5 - Posicionamento do Robô para manipulação de peças.

Desse modo, nota-se que se trata de um programa simples e que funciona de forma adequada as necessidades.

Um diagrama de blocos, com instruções mais detalhadas é mostrado abaixo:





Uma listagem deste programa inicial, que contém a idéia básica do programa final pode ser encontrada no ANEXO A.

6 CLP

6.1 Introdução

Os Controladores Lógicos Programáveis são os responsáveis pela realização do Controle, ou seja, o controle de SED (Sistema a Eventos Discretos). No controle de SED, as operações fundamentais (lógicas, aritméticas e temporização) são realizadas pelo CLP, que possui sinais de entrada e/ou saída que podem assumir valores discretos. Desse modo o CLP é a parte principal do sistema de controle.

A operação do CLP é baseada em um processamento cíclico conforme ilustrado na Figura 6.1. Em cada ciclo as entradas são coletadas, processadas e os resultados obtidos são enviados às saídas.

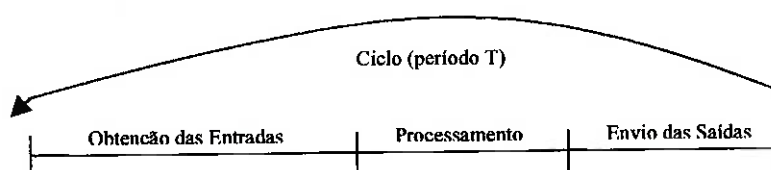


Figura 6.1 - Ciclo de Processamento do CLP.

6.2 Características do CLP a ser utilizado

O Controlador Lógico Programável que se apresentou disponível dentro do Departamento de Engenharia Mecânica foi o seguinte:

- Controlador de Média Complexidade marca Klockner Moeller tipo SUCOS PS 3-AC

Este controlador possui as seguintes características técnicas:

- Alimentação: 230/120V AC;
- 16 Entradas Digitais de 24V DC;
- 8 Saídas Digitais;
- 4 Entradas Analógicas (0-10V DC);
- 1 Saída Analógica (0-10V DC);
- 1 Entrada Para Contador (10kHz).

Um esquema representativo do controlador é mostrado na figura 6.2 a seguir:

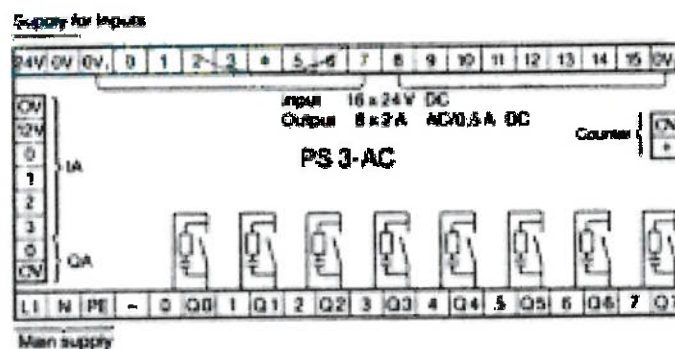


Figura 6.2 – Esquema do CLP utilizado

6.2.1 Características Técnicas das Entradas e Saídas Digitais

As características que devem ser ressaltadas para as Entradas e Saídas Digitais são apresentadas nas Tabelas 6.1 e 6.2.

Tabela 6.1 - Entradas Digitais.

Número de Entradas	16
Tipo	Acoplador óptico, técnica de ativação H para IEC 65
Configuração	2 Grupos
Faixa de Atuação	14-30V DC
Voltagem	0-7V DC
Faixa da Corrente de Atuação	9mA em 24V DC
On Delay (Low-High)	200 μ s aproximadamente
Off Delay (High-Low)	50 μ s aproximadamente
Comprimento do Cabo	400m
Indicador de Estado	LED vermelho

Tabela 6.2 - Saídas Digitais.

Número de Saídas	8
Tipo	Relé
Acoplador óptico	Possui
Configuração	8 Grupos
Faixa da Tensão de Saída	240V máx
Corrente de Saída	0.5 a 11A
Fator de Utilização	0.5
Frequência de Chaveamento por hora	10000
Corrente Residual a 120V/230V	1.5mA/3mA
Circuito Interno de Proteção	RC (470 Ω + 22 μ F)
Off Delay	6ms
Indicador de Estado	LED

6.3 Programação

A Programação inicial proposta para o CLP consiste de uma série de instruções individuais. Estas instruções são de 3 tipos:

- Informações de Entrada;
- Informações de Processamento;
- Informações de Saída.

Através dessas informações foi feito o programa inicial a ser carregado no CLP (conforme os métodos a seguir) com a finalidade de uma integração com o Robô ASEA, a nível de sinais de sincronismo. Da mesma forma que o programa no robô, este sofreu aprimoramentos ao longo do tempo para se adequar ao programa final implementado.

6.3.1 Meios de Programação

Todos os controladores das séries PS 3 podem ser programados de 3 formas:

- Através de um Microcomputador, utilizando listas de Instruções ou ainda graficamente através de um diagrama Ladder;
- Programação utilizando-se uma maleta de programação;
- Programação através de caneta óptica, que permite a geração de programas através de uma tela “VDU”.

A figura a seguir mostra estes três meios de programação:

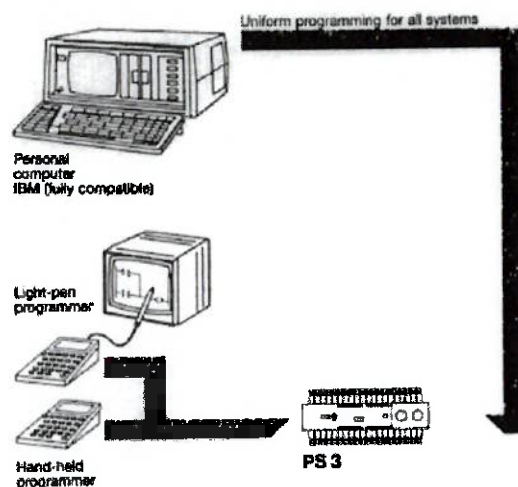


Figura 6.3 – Meios de programação do CLP

No caso deste projeto será executada a partir de um microcomputador utilizando-se lista de instruções (*Instruction List*)

A seguir é apresentado um exemplo de programa feito utilizando-se *Instruction List* e o seu respectivo diagrama de Circuito.

<u>Instrução</u>	<u>Descrição</u>
LI 0.0	Carrega o estado da entrada 0.0
AI 0.1	Realiza um AND lógico com o estado da entrada 0.1
=Q 0.0	De acordo com o resultado do AND, liga ou não a entrada 0.0

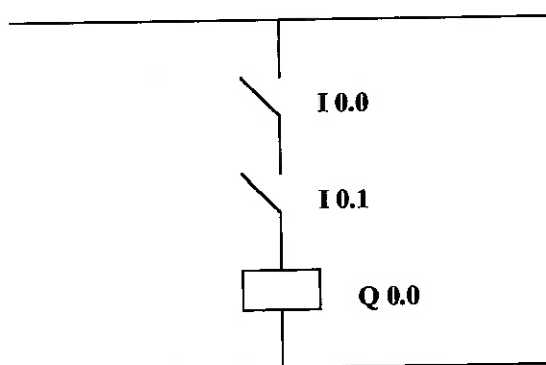


Figura 6.4 - Diagrama do Circuito do Exemplo.

O exemplo acima leva o valor da saída digital 0.0 para *High* ou *Low* (conforme o desejado) se e somente se os valores das entradas 0.0 e 0.1 forem iguais a *High*.

6.3.2 O Programa

O programa inicial que foi feito, conforme já mencionado diversas vezes, tem a finalidade de realizar um sincronismo das ações do Robô e portanto estará diretamente ligado ao programa inicial que foi feito para o Robô.

O programa possui os seguintes passos:

- Se valor da entrada I 0.0 = high e o valor das entradas I 0.1 e I 0.2 = low (recebeu número 01 em binário do Robô)
- Seta saída digital Q 0.1 = high e o valor das saídas Q 0.0 e Q 0.2 = low (envia número 02 em binário para o Robô)
- Se valor da entrada I 0.2 = high e o valor das entradas I 0.0 e I 0.1 = low (recebeu número 04 em binário do Robô)
- Seta saída digital Q 0.0 e Q 0.2 = high e o valor das saídas Q 0.1 = low (envia número 05 em binário para o Robô)
- Se valor da entrada I 0.1 e I 0.2 = high e o valor das entradas I 0.0 = low (recebeu número 06 em binário do Robô)
- Seta saída digital Q 0.0, Q 0.1 e Q 0.2 = low e Q 0.3 = high (envia número 08 em binário para o Robô)

Listagens do programa inicial que foi implementado no CLP, em *instruction List* e diagrama de relês são fornecidas respectivamente nos ANEXOS B e C.

7. Estudo do Funcionamento do Robô

Após ter-se terminado a primeira parte do trabalho de formatura, ou seja, concluídos os estudos iniciais de estudo de viabilidade do projeto, definição do projeto e estudo e documentação dos componentes do sistemas, partiu-se para um estudo do robô.

Este estudo foi auxiliado em grande parte pela “Aula Prática” dada pelo Prof. Jun Okamoto. Neste primeiro momento, aprendeu-se os comandos básicos de funcionamento do robô, como ligá-lo, como operá-lo com segurança e como fazer um programa utilizando a sua caixa de aprendizado.

7.1 Movimentação do Robô

Esta parte do estudo consistiu em se aprender os movimentos do robô, os vários tipos de sistemas de coordenadas que podem ser utilizados e as vantagens e movimentos proporcionados por cada um destes sistemas.

Vale aqui uma breve descrição destes sistemas de coordenadas:

- **Sistemas de coordenadas do Robô:** Este sistema foi o que mostrou-se mais prático, onde permite uma movimentação em coordenadas cilíndricas, permitindo rotação em torno do eixo da base, movimento na direção Z e outros 4 movimentos que podem ser visualizados na figura 7.1.

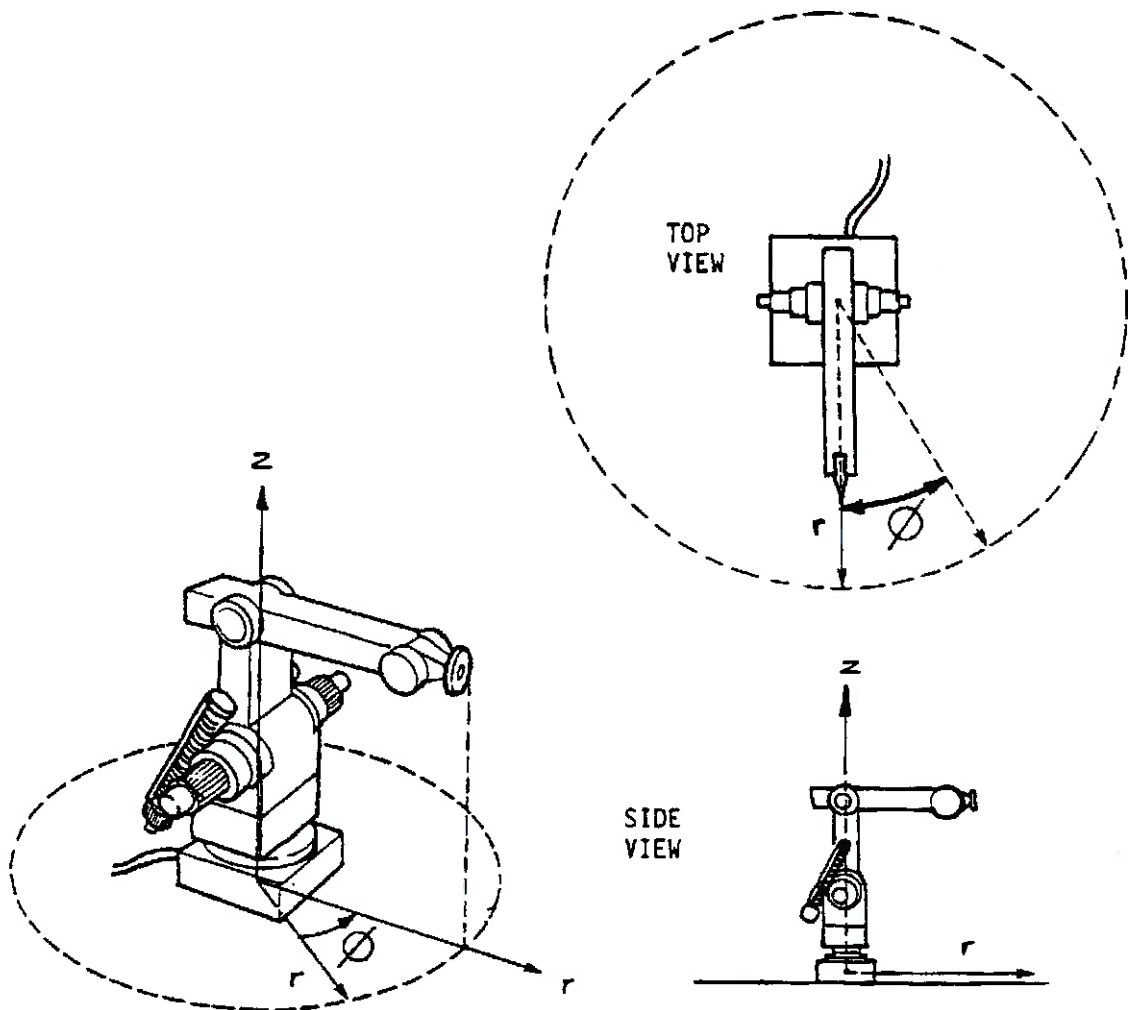


Figura 7.1 - Sistema de coordenadas cilíndricas do Robô.

Neste sistema o robô mexe basicamente uma articulação de cada vez, tornando mais fácil a previsão do seu movimento.

- **Sistema de coordenadas retangular:** Este sistema permite movimento segundo os eixos de coordenadas de um sistema cartesiano (X, Y, Z) e rotação em torno destes mesmos eixos. É bem útil quando se deseja mover-se de um ponto a outro segundo uma reta, pois o Robô moverá todas as articulações necessárias para fazer o movimento retilíneo.

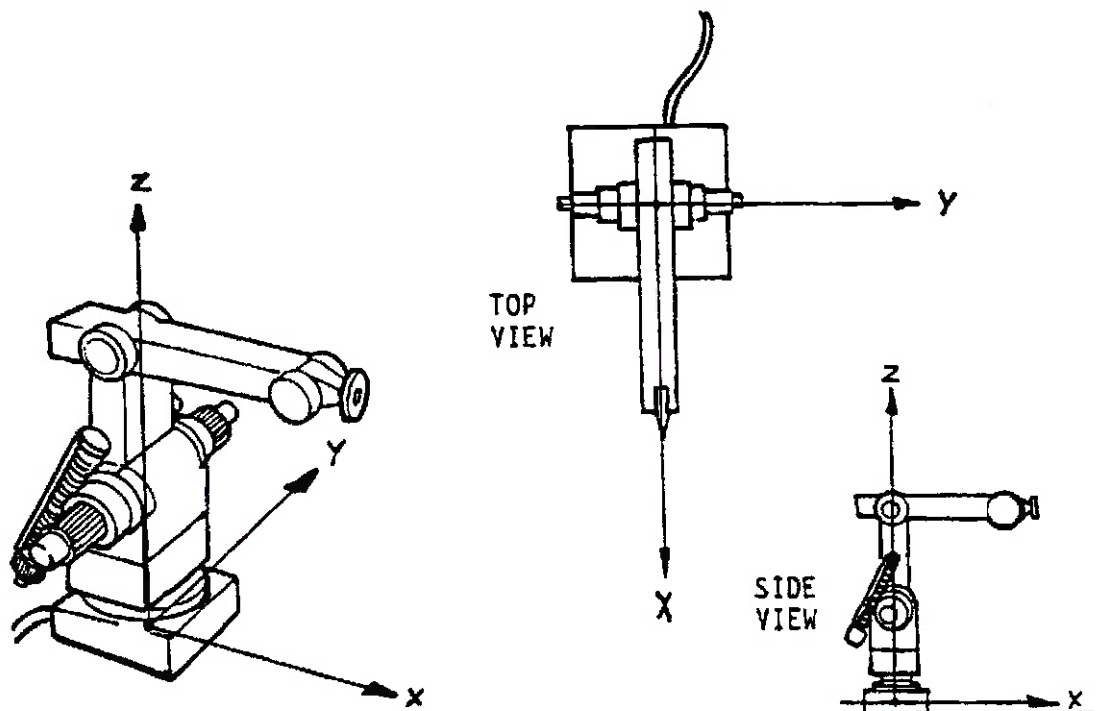


Figura 7.2 – Sistema de coordenadas retangular do Robô.

- **Sistema de coordenadas vetorial:** Este sistema de coordenadas move o robô segundo movimentos de translação e rotação (6 movimentos no total) em torno de um sistema de eixos disposto na sua garra.

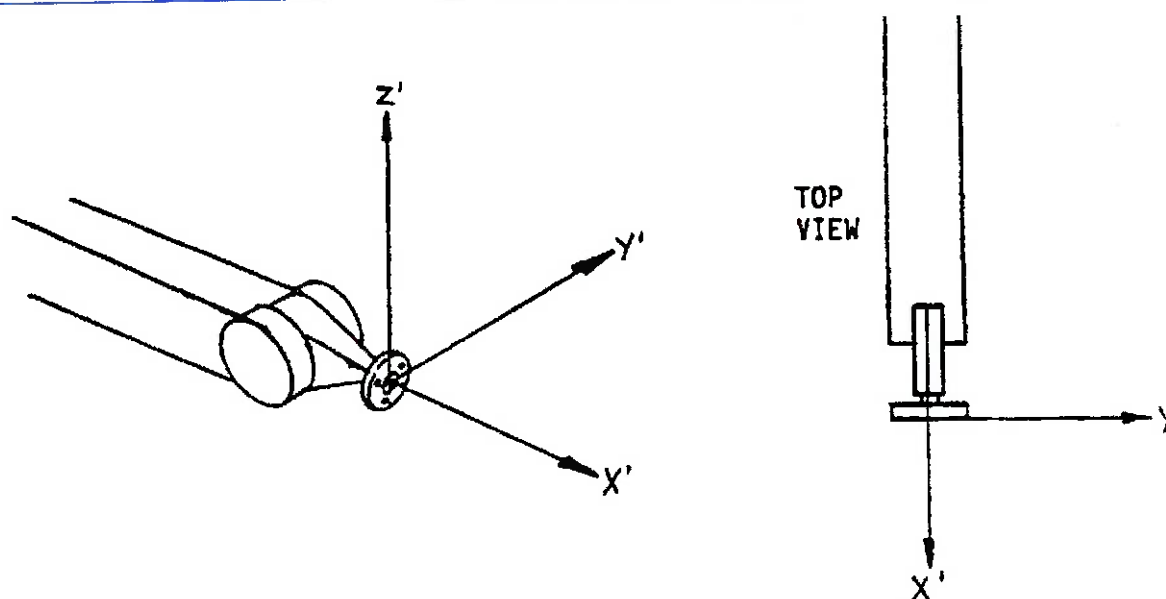


Figura 7.3 – Sistema de coordenadas da garra do Robô.

Obs:

- 1) É importante lembrar que cada sistema de coordenadas permite mover o Robô segundo seis graus de liberdade (6 modos de movimentação diferentes), bastando para isso utilizar o Joystick da caixa de aprendizado e a chave seletora que habilita três movimentos por vez no joystick.
- 2) Vale ainda lembrar que o caminho seguido pelo Robô durante a execução do programa não será necessariamente o mesmo que foi utilizado durante a sua programação. Isso só será obedecido se durante a movimentação entre dois pontos utilizando-se a caixa de aprendizado, seja selecionado o mesmo sistema de coordenadas que está programado no Robô durante estes comandos.

7.2 Estudo das Entradas e Saídas Digitais

O Robô disponível apresenta como entradas e saídas digitais 2 placas do modelo DSDX 110. Uma placa sendo a primária, que realiza algumas funções básicas internas do Robô (Motor ON/OFF, Movimentação da Garra, etc.) e uma secundária que também possui alguns bits ocupados para interrupções internas, porém tem-se boa parte livre para uso.

Essas placas se comunicam com o mundo externo através de 2 outras placas (que possuem bornes para conexões de fios com o mundo externo) que estão posicionadas na parte traseira da cabine de controle do Robô, sendo respectivamente as placas DSQC 124 (ligada à placa primária) e DSQC 122 (ligada à placa secundária). Um esquema da placa DSQC 122 é fornecido abaixo.

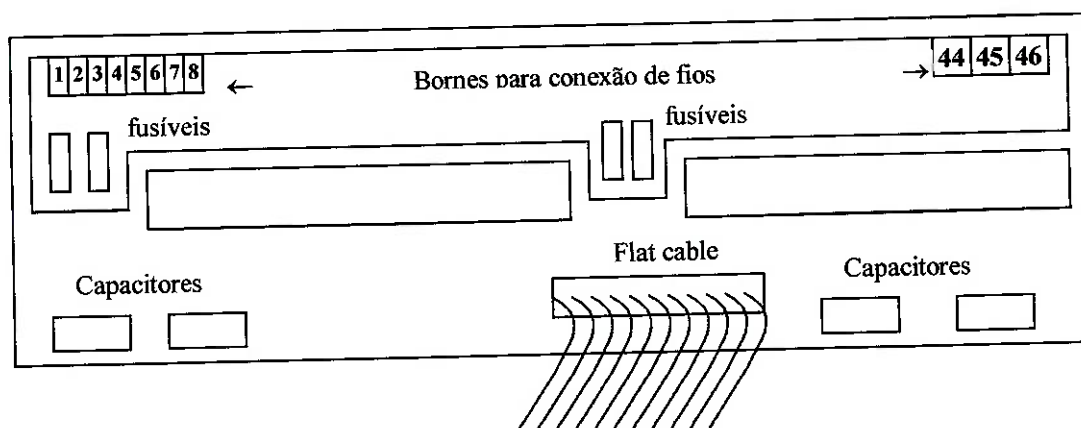


Figura 7.4 – Esquema da placa DSQC 122.

Inicialmente, usou-se somente a placa secundária de entradas e saídas digitais. Os quatro primeiros bits dessas entradas digitais são endereçados

pelo PORT nº 12 do Robô; já os quatro primeiros bits das saídas digitais são endereçados pelo PORT nº 2. Tabelas fornecendo os números dos bornes, e bits de entradas e saídas digitais são fornecidas abaixo.

Tabela 7.1 - Entradas digitais com seus respectivos bornes para a DSQC122.

Bit	0	1	2	3
INPUT	13	14	15	16
Borne	43	44	45	46

Tabela 7.2 - Saídas digitais com seus respectivos bornes para a DSQC 122.

Bit	0	1	2	3
OUTPUT	13	14	15	16
Borne	20	21	22	23

As indicações INPUT e OUTPUT referem-se aos LEDs que acendem no painel da cabine do Robô para um controle nas Entradas e Saídas ativas (HIGH) no momento.

Esta placa necessita de alimentação externa ou de uma outra fonte interna. No caso, utilizou-se a outra placa primária DSDX 110 como fonte.

Neste caso o esquema das ligações fica conforme mostrado na figura

7.5.

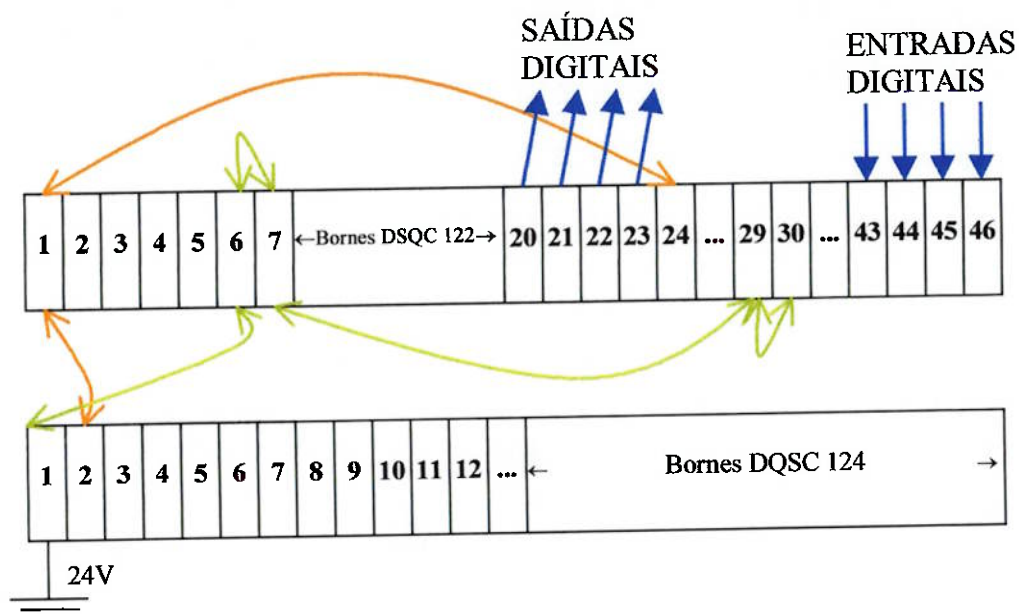


Figura 7.5 – Ligação entre as placas primária (DSQC 124) e secundária (DSQC 122) (entradas e saídas na secundária habilitadas).

Devido a problemas com parte desta placa, as saídas digitais da mesma não puderam mais ser utilizadas e passou-se a utilizar as saídas da placa primária.

As saídas da placa primária são endereçadas a partir do PORT nº 1 do Robô. Uma tabela com os dados de bits de saída e os respectivos bornes é mostrada a seguir.

Tabela 7.3 - Saídas digitais com seus respectivos bornes para a DSQC 124.

Bit	0	1	2	3
OUTPUT	3	4	5	6
Borne	8	9	10	11

Nesta nova situação, o esquema de ligações com alimentação entre as placas fica conforme mostrado abaixo.

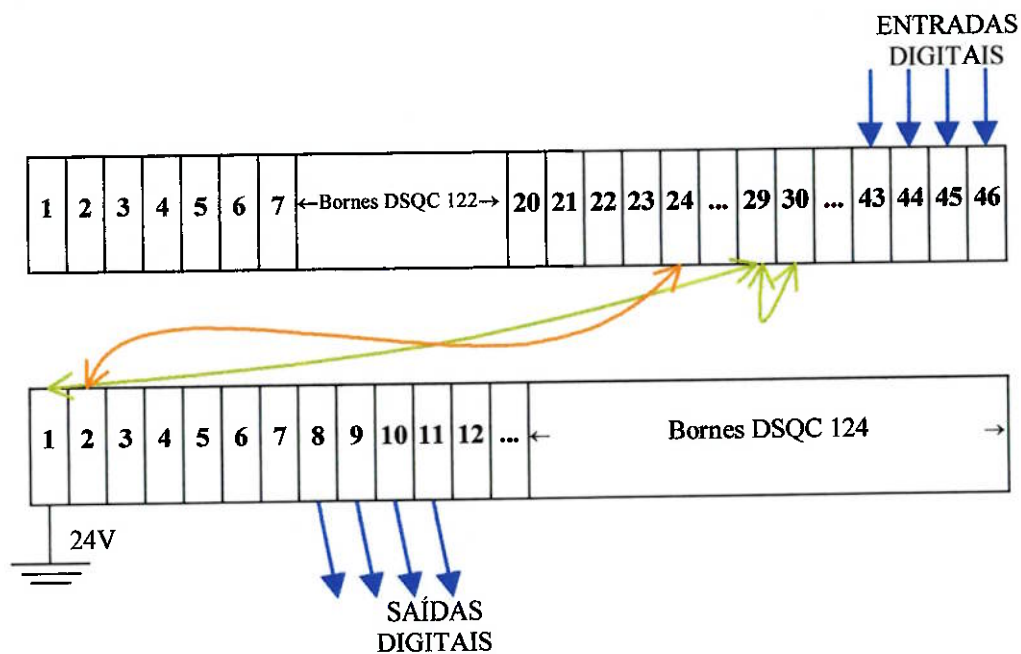


Figura 7.6 – Ligação entre as placas primária (DSQC 124) e secundária (DSQC 122) (saídas da placa secundária desabilitadas).

Um esquema das placas DSDX 110 e DSQC122 pode ser encontrado no **ANEXO D**.

8. Estudo do Funcionamento do CLP

8.1 Introdução

No estudo de funcionamento do CLP, aprendeu-se basicamente a utilização do software específico para a sua programação, as características destes CLP (que já estavam listadas na primeira parte do trabalho) e também a trabalhar com suas entradas e saídas digitais.

O jeito mais prático encontrado para aprender sua utilização foi através da leitura dos manuais e elaboração de pequenos programas a fim de se descobrir os vários recursos existentes no software.

8.2 Descrição da Programação do CLP e suas saídas digitais

Para a programação no CLP existem três meios de programação:

- **Programação LDI:** Consiste na elaboração de programa utilizando comandos em Instruction List. Este foi o método utilizado para elaboração dos programas, e portanto exemplos de programação com este método podem ser vistos mais adiante.
- **Programação DDC:** Consiste na elaboração de programas utilizando-se diagramas de relés. Um exemplo de programa de programação pode ser visto abaixo.

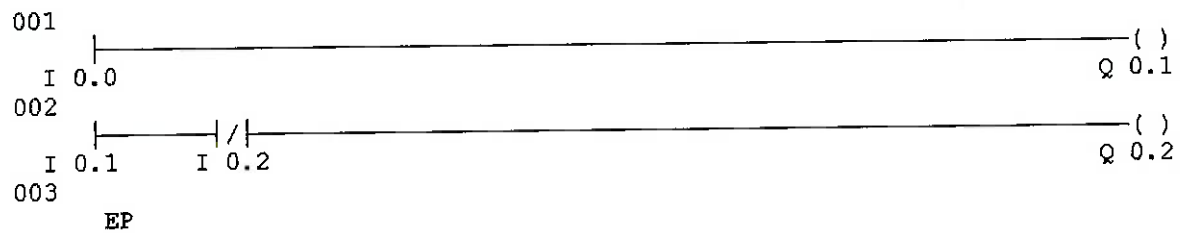


Figura 8.1 – Diagrama de relés caracterizando a programação DDC.

- **Programação DDB:** Consiste na programação utilizando-se diagramas com lógica do tipo e, ou etc.. Um exemplo deste tipo de programação pode ser visto a seguir.

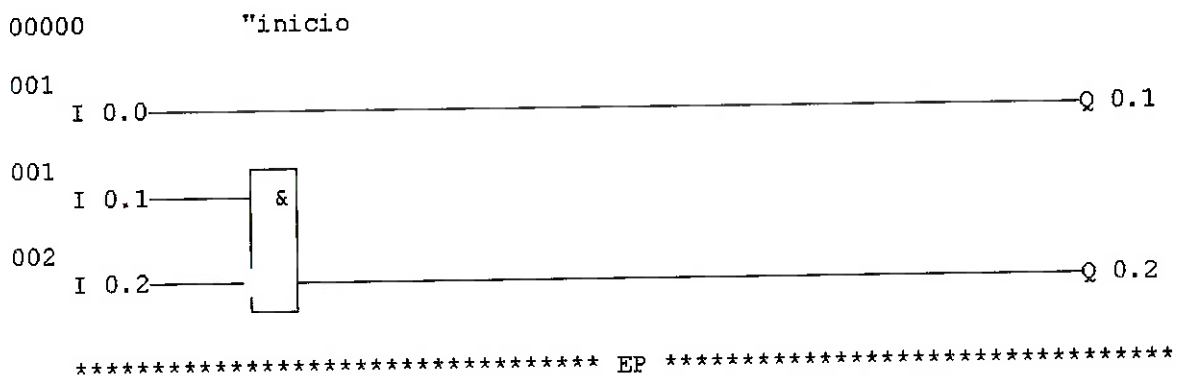


Figura 8.2 – Diagrama característico de uma Programação DDB.

As entradas e saídas digitais do CLP são acessadas por meio de comandos como SET, RESET, ou =<expressão lógica>. O meio utilizado para programação foram as expressões lógicas, por meio de diagramas de relé. Para um maior detalhamento vide os exemplos do item 9 abaixo ou ainda o manual de instrução do CLP.

As entradas físicas do CLP estão divididas em 2 grupos de 8 entradas cada, com terra próprio para cada grupo conforme ilustrado na figura 8.3.

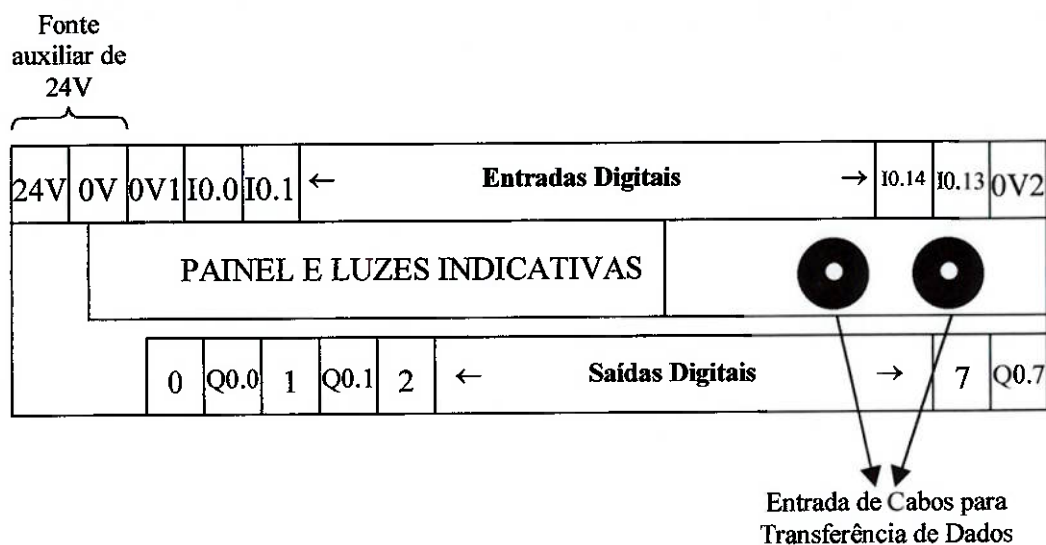


Figura 8.3 – Esquema representativo do CLP (entradas e saídas digitais).

As saídas do CLP são apresentadas do tipo relé (contato seco), necessitando portanto de alimentação para que ao fechadas (de acordo com os sinais enviados e utilizados no programa) possam enviar um sinal de 24V nessa saída. Um esquema que ilustra essa descrição é mostrado na figura 8.4 a seguir.

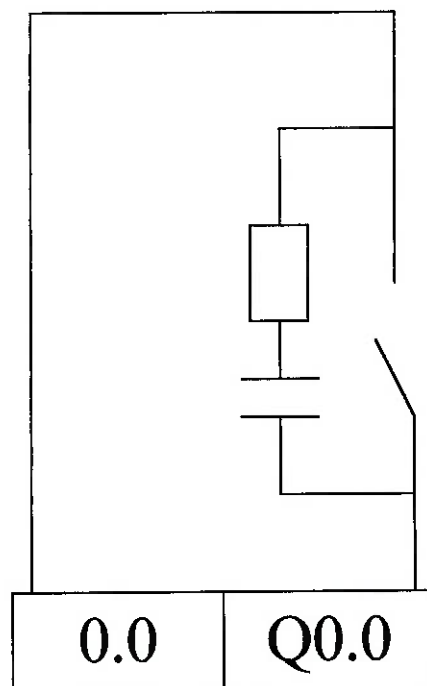


Figura 8.4 – Esquema das saídas do CLP.

9. Realização de Testes no CLP

Os seguintes programas de testes foram utilizados para testes iniciais no PLC, visando testes de comunicação PLC ↔ Micro, Entradas e Saídas digitais no PLC, além de medição destas entradas e saídas.

Programa “TESTE 1”

00000	"INICIO	
001	L I 0.0	LE ENTRADA I0
002	A I 0.1	AND COM ENTRADA I1
003	= Q 0.1	SETA SAIDA Q1 CASO SATISFEITAS
004	EP	

Programa “TESTE 2”

00000	"INICIO	
001	LN I 0.0	LE OPOSTO DA ENTRADA I0
002	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
003	= Q 0.0	SETA SAIDA Q0 CASO SATISFEITAS
004	EP	

Programa “TESTE 3”

00000	"INICIO	
001	LN I 0.0	LE OPOSTO DA ENTRADA I0
002	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
003	= Q 0.0	SETA SAIDA Q0 CASO SATISFEITAS
004	= Q 0.1	SETA SAIDA Q1 CASO SATISFEITAS
005	= Q 0.2	SETA SAIDA Q2 CASO SATISFEITAS
006	= Q 0.3	SETA SAIDA Q3 CASO SATISFEITAS
007	EP	

Programa “TESTE 4”

00000	"inicio	
001	L I 0.0	LE ENTRADA I0
002	= Q 0.1	SETA SAIDA Q1 CASO I0=HIGH
003	L I 0.1	LE ENTRADA I1
004	AN I 0.2	AND COM OPOSTO DA ENTRADA I2
005	= Q 0.2	SETA SAIDA Q2 CASO SATISFEITAS
006	EP	

Programa “TESTE 5”

00000	"inicio	
001	L I 0.0	LE ENTRADA I0
002	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
003	AN I 0.2	AND COM OPOSTO DA ENTRADA I2
004	= Q 0.1	SETA SAIDA Q1 CASO SATISFEITAS
005	EP	

Programa “TESTE 6”

00000	"inicio	
001	L I 0.0	LE ENTRADA I0
002	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
003	AN I 0.2	AND COM OPOSTO DA ENTRADA I2
004	= Q 0.1	SETA SAIDA Q1 CASO SATISFEITAS
005	LN I 0.0	LE OPOSTO DA ENTRADA I0
006	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
007	A I 0.2	AND COM ENTRADA I2
008	= Q 0.0	SETA SAIDA Q0 CASO SATISFEITAS
009	= Q 0.2	SETA SAIDA Q2 CASO SATISFEITAS
010	LN I 0.0	LE OPOSTO DA ENTRADA I0
011	A I 0.1	AND COM ENTRADA I1
012	A I 0.2	AND COM ENTRADA I2
013	= Q 0.0	SETA SAIDA Q0 CASO SATISFEITAS
014	EP	

Obs: Com este programa, verificou-se que a saída Q 0.0 sempre vai ficar em “LOW” (nível zero) pois o CLP entende que uma das linhas manda ela ir para HIGH e a outra para LOW. Portanto, o programa que pensava-se utilizar na primeira parte do trabalho de formatura (representado pelo programa Teste 7) não funcionaria conforme o imaginado anteriormente.

Programa “TESTE 7”

Programa que inicialmente pensou-se em utilizar no robô, mas não funcionaria do modo esperado (devido ao que foi citado na Obs acima) e portanto foi substituído pelo programa “TESTE 8”.

00000	"Inicio	
001	L I 0.0	LE ENTRADA I0
002	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
003	AN I 0.2	AND COM OPOSTO DA ENTRADA I2
004	= Q 0.1	SETA SAIDA Q1 CASO SATISFEITAS
005	LN I 0.0	LE OPOSTO DA ENTRADA I0
006	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
007	A I 0.2	AND COM ENTRADA I2
008	= Q 0.0	SETA SAIDA Q0 CASO SATISFEITAS
009	= Q 0.2	SETA SAIDA Q2 CASO SATISFEITAS
010	LN I 0.0	LE OPOSTO DA ENTRADA I0
011	A I 0.1	AND COM ENTRADA I1
012	A I 0.2	AND COM ENTRADA I2

013	= Q 0.0	SETA SAIDA Q0 CASO SATISFEITAS
014	= Q 0.1	SETA SAIDA Q1 CASO SATISFEITAS
015	= Q 0.2	SETA SAIDA Q2 CASO SATISFEITAS
016	EP	

Programa "TESTE 8"

Programa que inicialmente será utilizado no robô

00	"Inicio	
001	L I 0.0	LE ENTRADA I0
002	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
003	AN I 0.2	AND COM OPOSTO DA ENTRADA I2
004	= Q 0.1	SETA SAIDA Q1 CASO SATISFEITAS
005	LN I 0.0	LE OPOSTO DA ENTRADA I0
006	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
007	A I 0.2	AND COM ENTRADA I2
008	= Q 0.0	SETA SAIDA Q0 CASO SATISFEITAS
009	= Q 0.2	SETA SAIDA Q2 CASO SATISFEITAS
010	LN I 0.0	LE OPOSTO DA ENTRADA I0
011	A I 0.1	AND COM ENTRADA I1
012	A I 0.2	AND COM ENTRADA I2
013	= Q 0.3	SETA SAIDA Q3 CASO SATISFEITAS
014	EP	

Programa “TESTE 24”

Este programa foi feito para testar as entradas e saídas digitais, ou seja a ligação com 24 V, mais especificamente.

00000	"Inicio	
001	LN I 0.0	LE OPOSTO DA ENTRADA I0
002	AN I 0.1	AND COM OPOSTO DA ENTRADA I1
003	= Q 0.0	SETA SAIDA Q0 CASO SATISFEITAS
004	L I 0.3	LE ENTRADA I3
005	= Q 0.1	SETA SAIDA Q1 CASO SATISFEITAS
006	L I 0.4	LE ENTRADA I4
007	= Q 0.2	SETA SAIDA Q2 CASO SATISFEITAS
008	EP	

Com este programa, alimentou-se as entradas digitais do CLP com as suas próprias saídas a fim de verificar a sua tensão e validade.

10. Realização de testes no Robô

Após o estudo inicial do robô, foram feitos alguns programas nos moldes do programa final, ou seja, visando reproduzir a mesma arquitetura e comandos básicos deste, porém, programas mais simplificados.

Nestes programas, não foram jogadas entradas e saídas digitais do CLP e estes valores foram supostos recebidos, ou seja, foram feitas simulações e nesta primeira etapa foram supostos “recebidos” os valores do CLP.

O funcionamento básico destes programas é o mesmo, porém com algumas peculiaridades entre um e outro. Todos eles simulam um posicionamento de peças (Pick-and-Place) em que o robô manda valores para as saídas digitais entre um posicionamento e outro a fim de simular uma comunicação com o CLP.

Programa 1:

10: V = 200MM/S MAX = 500MM/S

20: ROBOT COORD

30: TCP 0

40: FRAME 0

50: LET R5 = 7

60: JUMP TO 100 IF R5 <> 7

70: POS V = 50% FINE L

```
80: LET R4 = 1
90: TRANSF R4 TO PORT Nº 1
100: WAIT 5S
110: LET R5 = 2
120: JUMP TO 140 IF R5 <> 2
130: GRIPPER WAIT 0,5S
140: JUMP TO 160 IF R6 <> 3
150: POS V = 50% FINE L
160: LET R4 = 4
170: TRANSF R4 TO PORT Nº 1
180: WAIT 2S
190: LET R5 = 5
200: JUMP TO 220 IF R5 <> 5
210: GRIPPER WAIT 3S
220: LET R4 = 6
230: TRANSF R4 TO PORT Nº 1
240: RETURN
```

Programa 2:

```
10: V = 200MM/S MAX = 500MM/S
20: ROBOT COORD
30: TCP 0
40: FRAME 0
50: LET R5 = 7
```

```
60: JUMP TO 100 IF R5 <> 7
70: POS V = 50% FINE L
80: LET R4 = 1
90: TRANSF R4 TO PORT Nº 1
100: WAIT 5S
110: LET R5 = 2
120: JUMP TO 140 IF R5 <> 2
130: GRIPPER WAIT 0,5S
135: LET R6 = 3
140: JUMP TO 160 IF R6 <> 3
150: POS V = 50% FINE L
160: LET R4 = 4
170: TRANSF R4 TO PORT Nº 1
180: WAIT 2S
190: LET R5 = 5
200: JUMP TO 220 IF R5 <> 5
210: GRIPPER WAIT 3S
220: LET R4 = 6
230: TRANSF R4 TO PORT Nº 1
240: RETURN
```

Obs.: No programa 1 (não existe a instrução de nº 135 e portanto a instrução do comando JUMP não é executada) o robô realiza um posicionamento a menos que no programa 2 e este fato vai servir para testar a comunicação com o CLP.

Programa 3:

10: V = 300MM/S MAX = 800MM/S

20: ROBOT COORD

30: TCP 0

40: FRAME 0

50: POS V = 50% FINE L

60: LET R5 = 7

70: JUMP TO 110 IF R5 <> 7

80: POS V = 50% FINE L

90: POS V = 50% FINE L

100: POS V = 50% FINE L

110: LET R4 = 1

120: TRANSF R4 TO PORT N° 1

130: WAIT 5S

140: LET R5 = 2

150: JUMP TO 170 IF R5 <> 2

160: GRIPPER WAIT 2.0S

170: LET R6 = 3

180: JUMP TO 240 IF R6 <> 3

190: POS V = 50% FINE L

200: POS V = 50% FINE L

210: POS V = 50% FINE L

220: POS V = 50% FINE L

230: POS V = 50% FINE L

240: LET R4 = 4
250: TRANSF R4 TO PORT Nº 1
260: WAIT 2S
270: LET R5 = 5
280: JUMP TO 220 IF R5 <> 5
290: GRIPPER WAIT 3S
300: LET R4 = 6
310: TRANSF R4 TO PORT Nº 1
320: RETURN

Obs.: Verificou-se que para mover-se entre 2 pontos distantes (opostas em relação ao robô) é preciso colocar outros pontos no meio pois senão ocorre travamento do robô em algumas posições.

11. Programa final de comunicação Robô – CLP

11.1 Descrição dos programas

Os programas finais, implementados no CLP e Robô, foram feitos visando manter a idéia inicial da célula de manufatura. Apesar de não se ter uma célula de manufatura realmente, o programa visa aplicações práticas na indústria, que no futuro possam ser empregadas no desenvolvimento da célula de manufatura.

A idéia inicial de se fazer um posicionamento de peça dentro do torno, não foi possível de ser concretizada por alguns motivos, são eles:

- O torno (CNC) disponível não possuía uma abertura de porta automática, nem uma placa de entradas e saídas digitais para comunicação externa, o que impossibilitou sua utilização no trabalho. Vale ressaltar, que esta placa é opcional no CNC e pode ser adquirida no futuro.

- As posição que o robô se encontrava em relação ao torno, fazia com que a entrada da sua garra no torno com uma peça se torna-se bastante complicada, sendo que como o robô estava sendo sendo utilizado em outros projetos, a posição do robô em relação ao torno poderia variar e ocasionar algum acidente.

- A garra utilizada pelo robô não é ideal para se pegar peças cilíndricas e se rotacionar elas, pois muitas vezes elas acabam rodando em relação à garra e não se garantindo sua posição final.

Portanto, pelos motivos expostos acima, optou-se por medida de segurança não se utilizar um posicionamento no torno, porém outro tipo de teste, cujo efeito prático fosse o mesmo.

O programa final, visa simular uma célula de produção de algum tipo de peças, e um esquema do teste final realizado pode ser observado na figura 11.1.

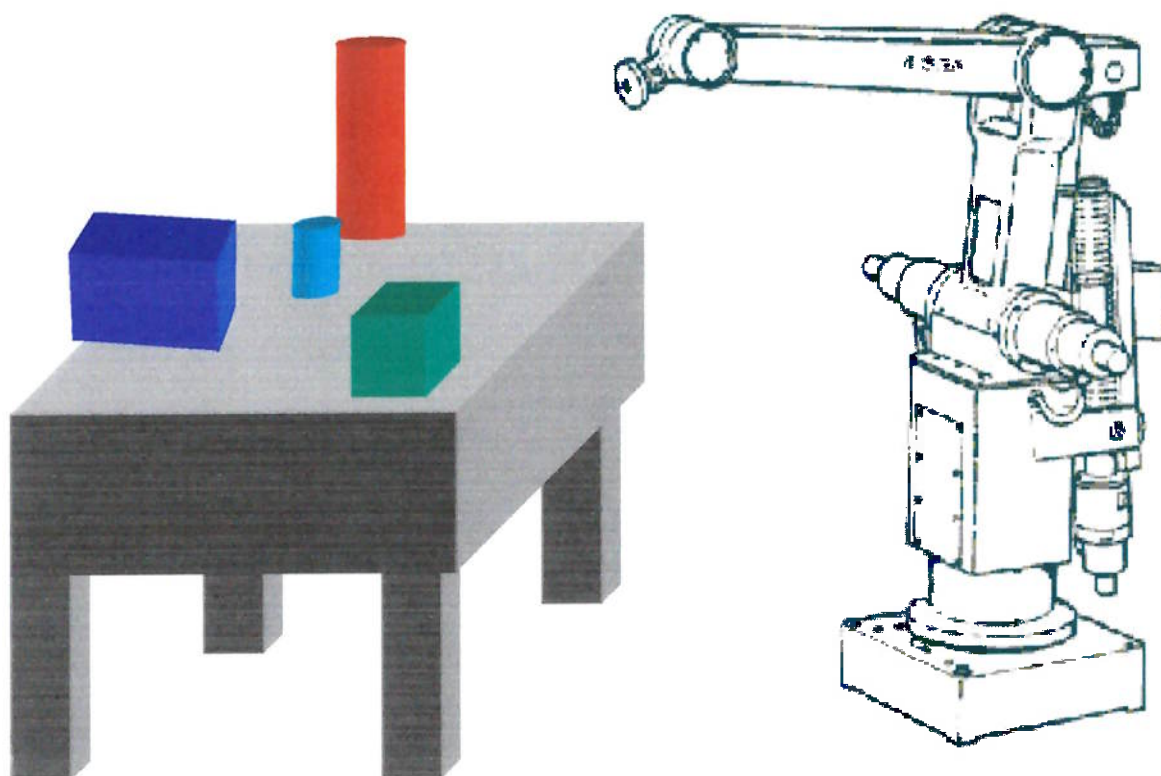


Figura 11.1 - Esquema da situação do teste final.

O programa visa contemplar a seguinte situação:

- O robô, após receber um sinal de controle, vai para uma posição (representada pelo poste vermelho da figura) onde realizará uma rotina de checagem de posicionamento em relação à mesa;
- Após ter realizado esta checagem, o robô avisa o CLP e espera por um novo sinal de comando deste. Além do sinal recebido do CLP, ele aguarda um sinal vindo da botoeira implementada. O sinal da botoeira objetiva representar em uma situação real, por exemplo, um sinal enviado por um sensor de presença de peças em uma célula de fabricação. Após receber este sinal, o robô posiciona-se sobre a peça, agarrando-a e avisando o fato ao CLP.
- Após pegar a peça ele aguarda um sinal de disponibilidade de máquina (no caso de uma CNC disponível para ele colocar a peça). Recebendo o sinal, o robô posiciona a peça no CNC, no caso representado pela plataforma verde, e sai da máquina para realizar outra operação ou aguardar até que a peça fique pronta.
- Recebendo o sinal de que a peça ficou pronta (o sinal seria enviado pelo CNC, mas no nosso caso foi implementado através da botoeira). O CLP avisa o robô e ordena que este retire a peça do torno e a coloque no Rack de peças prontas. Após colocar a peça no Rack, o robô avisa o CLP da operação concluída.

- Feito isso, volta-se para uma situação inicial, com execução da checagem inicial de posições que não é mais realizada, e o robô ficaria pegando peças e posicionando-as no CNC para serem usinadas e guardando as peças usinadas no Rack.

Vale lembrar que os programas do CLP e robô ficam rodando em um loop infinito, até que por exemplo, algum botão apertado mande-os executar alguma tarefa.parar.

A) Programa no Robô.

Para realizar as ações acima, o programa que foi implementado no robô possui a seguinte sequência de passos:

- Após a inicialização, envia sinal para o CLP;
- Se sinal recebido do CLP = 1 então
 - Move-se para posição de checagem de posição (poste vermelho);
 - Avisa CLP ao fim da movimentação;
 - Seta registrador para não entrar neste loop novamente;
- Se sinal recebido do CLP = 2 então
 - Move-se para pegar a peça;
 - Avisa CLP que pegou peça;

-
- Seta registrador para entrar neste loop somente no momento exato de pegar nova peça;
 - Se sinal recebido do CLP = 4 então
 - Posiciona peça no CNC;
 - Sai do CNC;
 - Avisa CLP do posicionamento da peça;
 - Seta registrador para entrar neste loop somente no momento exato de posicionar nova peça;
 - Se sinal recebido do CLP = 8 então
 - Retira peça do CNC;
 - Coloca peça no Rack;
 - Avisa CLP do final da operação;
 - Seta registrador para entrar neste loop somente no momento exato de retirar nova peça do torno;

B) Programa do CLP.

- Se sinal recebido do Robô for = 1 e receber sinal da botoeira Verde então
 - Envia sinal para robô realizar sincronismo de posição.
- Se sinal recebido do Robô for = 2 e receber sinal da botoeira Azul então
 - Envia sinal para robô pegar peça.

- Se sinal recebido do Robô for = 4 e receber sinal da botoeira Azul então
 - Envia sinal para robô posicionar peça no CNC.
- Se sinal recebido do Robô for = 8 e receber sinal da botoeira Azul então
 - Envia sinal para robô retirar peça do CNC e posicionar no Rack.

Vale aqui ressaltar, que foram implementados alguns temporizadores para os botões da botoeira. Esses temporizadores, visam aproximar a situação de um caso real, onde por exemplo exista um sensor de presença de peças, e portanto, após o robô agarrar a peça e retirá-la do dispositivo onde está o sensor, o sinal enviado por esse passa a ser zero, e portanto os programas implementados tanto no CLP quanto no robô devem estar atentos a este fato. Com esse objetivo, implementou-se no programa do CLP alguns temporizadores que mantêm o sinal da botoeira em High por certo tempo após apertado o botão e depois levam este sinal de volta para LOW.

Uma listagem dos programas implementados no CLP e Robô são fornecidas nos itens seguintes.

11.2 Programa final do CLP

Programa 351 do PLC (utilizado com o programa 35 do robô)

```
00000      "inicio
001         TR0
002         [ ] S:  'SAIDA0
003         [ ] STOP:
004         [w] I:   KW30
005         [ ] EQ:  M 0.0
006         TR1
007         [ ] S:  'SAIDA1
008         [ ] STOP:
009         [w] I:   KW30
010         [ ] EQ:  M 0.1
011         TR2
012         [ ] S:  'SAIDA2
013         [ ] STOP:
014         [w] I:   KW30
015         [ ] EQ:  M 0.2
016         TR3
017         [ ] S:  'SAIDA3
018         [ ] STOP:
019         [w] I:   KW30
020         [ ] EQ:  M 0.3
```

021	L 'ENT0	Entrada 1
022	AN 'ENT1	Entrada 1
023	AN 'ENT2	Entrada 2
024	A 'VERDE	Botao verde
025	L 'SAIDA0	
026	AN M 0.0	
027	O	
028	= 'SAIDA0	
029	LN 'ENT0	Entrada 1
030	A 'ENT1	Entrada 1
031	AN 'ENT2	Entrada 2
032	A 'AZUL	Botao azul
033	L 'SAIDA1	
034	AN M 0.1	
035	O	
036	= 'SAIDA1	
037	L 'ENT0	Entrada 1
038	A 'ENT1	Entrada 1
039	AN 'ENT2	Entrada 2
040	A 'AZUL	Botao azul
041	L 'SAIDA2	
042	AN M 0.2	
043	O	
044	= 'SAIDA2	
045	LN 'ENT0	Entrada 1

046	AN 'ENT1	Entrada 1
047	A 'ENT2	Entrada 2
048	A 'AZUL	Botao azul
049	L 'SAIDA3	
050	AN M 0.3	
051	O	
052	= 'SAIDA3	
53	EP	

11.3 Programa final do Robô

10: ROBOT COORD

20: V=200MM/S MÁX=500MM/S

30: TCP 0

40: FRAME 0

50: LET R5 = 0

60: LET R4 = 0

70: LET R6 = 1

80: TRANSF R6 TO PORT Nº 1

90: JUMP TO 230 IF R4 < > 1

100: JUMP TO 230 IF R5 = 1

110: POS V=100% FINE XL

120: GRIPPER WAIT 1.5 S

130: RECT COORD

140: POS V=25% FINE

150: POS V=25% FINE
160: POS V=25% FINE
170: POS V=25% FINE
180: POS V=25% FINE
190: POS V=25% FINE
200: LET R5 = 1
210: LET R6 = 2
220: TRANSF R6 TO PORT 1
230: JUMP TO 330 IF R4 < > 2
240: JUMP TO 330 IF R5 = 2
250: POS V=100% FINE XL
260: POS V=75% FINE XL
270: POS V=30% FINE
280: POS V=30% FINE
290: GRIPPER WAIT 1.5 S
300: LET R5 = 2
310: LET R6 = 3
320: TRANSF R6 TO PORT Nº 1
330: JUMP TO 440 IF R4 < > 4
340: JUMP TO 440 IF R5=3
350: POS V=30% FINE
360: POS V=100% FINE
370: POS V=30% FINE
380: GRIPPER WAIT 1.5 S
390: POS V=30% FINE

400: POS V=80% FINE L
410: LET R5 = 3
420: LET R6 = 4
430: TRANSF R6 TO PORT Nº 1
440: JUMP TO 560 IF R4 < > 8
450: JUMP TO 560 IF R5 = 4
460: POS V= 80% FINE L
470: POS V=30% FINE
480: GRIPPER WAIT 1.5 S
490: POS V=30% FINE
500: POS V=80%
510: POS V=30% FINE
520: GRIPPER WAIT 1.5 S
530: POS V=30% FINE
540: POS V=80% FINE
550: LET R5 = 4
551: LET R6 = 2
552: TRANSF R6 TO PORT Nº 1
560: FETCH R4 FROM PORT Nº 12
570: JUMP TO 90 IF R4 = 1
580: JUMP TO 230 IF R4 = 2
590: JUMP TO 330 IF R4 = 4
600: JUMP TO 70 IF R4 = 8
610: JUMP TO 90 IF R4 = 0
620: RETURN

12. Considerações Finais

O presente trabalho apresentou um estudo sobre a interligação entre um Robô ASEA e um CLP. Também foi mostrado no decorrer do texto outras alternativas para comunicação, como por exemplo as redes ASi e PROFIBUS e também a sua inviabilidade para o contexto atual deste projeto.

A comunicação entre Robô e CLP possui aplicações muito práticas, como por exemplo uma “mini-célula” de fabricação de certo tipo de peça onde as botoeiras utilizadas poderiam ser facilmente substituídas por algum tipo de sensor de presença de peça ou outro que se encaixe no tipo de aplicação.

O que foi implementado está longe de ser uma célula de manufatura, entretanto pode ser considerado um passo importante na sua implementação como a comunicação entre diversos equipamentos, gerenciada pelo CLP. Um exemplo disto seria uma célula em que se tivesse um robô pegaria determinada peça e posicionaria a mesma num CNC, e, enquanto aguarda a usinagem dessa peça poderia executar outra função como o abastecimento de uma Fresa.

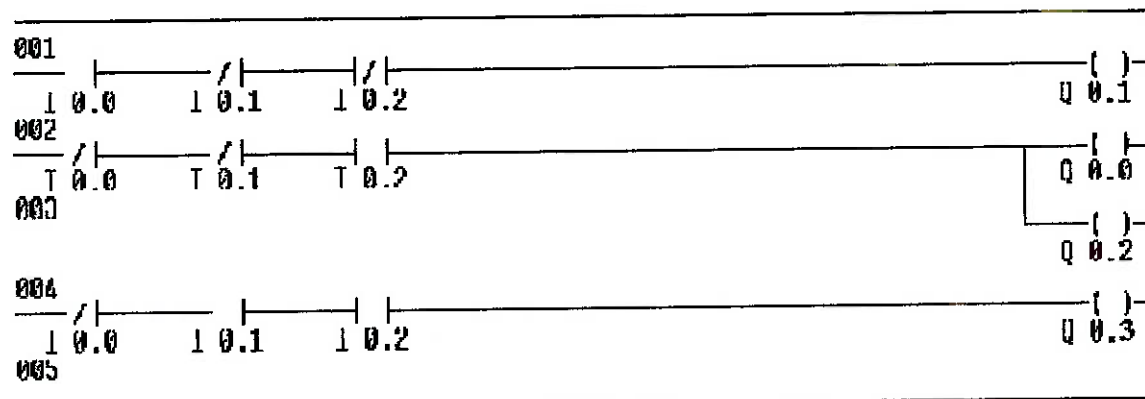
Por fim, vale ressaltar, o possível relacionamento com outros trabalhos desenvolvidos por outros alunos, como o da comunicação serial com o robô e o do controlador da guia de movimentação sobre a qual está apoiado o robô. Com a união destes trabalhos pode-se começar a idealizar uma célula de manufatura maior, onde se existiria uma maior flexibilidade de atividades.

ANEXO A – Listagem do Programa inicial no Robô

10:	V = 50 MM/S MAX = 100 MM/S	{ seta velocidades básica e máx.}
20:	GET TO R5 FROM PORT "Y"	{ armazena entrada no R5}
30:	JUMP TO 70 IF R5 <> 8	{ Jump condicional}
40:	POS V = 50%	{ move-se para posição M1}
50:	SET R4 = 1	{seta registrador R4 com valor "1"}
60:	TRANSFER R4 TO PORT "X"	{ transfere valor de R4 para saída}
70:	WAIT 5.0 S	{ espera 5 segundos}
80:	GET TO R5 FROM PORT "Y"	{ armazena entrada no R5}
90:	JUMP TO 110 IF R5 <> 2	{ Jump condicional}
100:	PEGA OBJETO	{ Pegar objeto}
110:	JUMP TO 130 IF R6 <> 3	{ Jump condicional}
120:	POS V = 50%	{ move-se para posição M2}
130:	SET R4 = 4	{seta registrador R4 com valor "4"}
140:	TRANSFER R4 TO PORT "X"	{ transfere valor de R4 para saída}
150:	WAIT 5.0 S	{ espera 5 segundos}
160:	GET TO R5 FROM PORT "Y"	{ armazena entrada no R5}
170:	JUMP TO 190 IF R5 <> 5	{ Jump condicional}
180:	SOLTAR OBJETO	{ Soltar objeto}
190:	SET R4 = 6	{seta registrador R4 com valor "6"}
200:	TRANSFER R4 TO PORT "X"	{ transfere valor de R4 para saída}
210:	RETURN	{volta ao início do programa}

ANEXO B – Listagem em *Instruction List* do Programa inicial no CLP

00000	“Inicio
001	L I 0.0
002	AN I 0.1
003	NA I 0.2
004	=Q 0.1
005	LN I 0.0
006	NA I 0.1
007	A I 0.2
008	= Q 0.0
009	= Q 0.2
010	LN I 0.0
011	A I 0.1
012	A I 0.2
013	= Q 0.3
014	EP

ANEXO C – Diagrama de Relés do programa inicial no CLP

ANEXO D – Diagrama das Placas do Robô

Bibliografia

- [1] ACTUATOR SENSOR INTERFACE – Catálogo Siemens – 1997.
- [2] www.as-interface.com
- [3] POPP MANFRED, **The rapid Way to PROFIBUS – DP.**
- [4] SIMATIC - **Components for Totally Integrated Automation** - Catalog ST70 – 1997 / Siemens.
- [5] www.profibus.com
- [6] ASEA Industrial Robots – **Product Manual**
- [7] ASEA Industrial Robots – **Programming Manual**
- [8] MIYAGI, Paulo E.. **Controle programável.** São Paulo, Edgard Blücher, 1996.
- [9] ELECTRONIC CONTROL SYSTEM – **Hardware and Design Engineering** – ElettroncaVeneta.
- [10] ELECTRONIC CONTROL SYSTEM – **Software and Programming Language** - ElettroncaVeneta.
- [11] ELECTRONIC CONTROL SYSTEM – **Operating Manual** -ElettroncaVeneta.