RAFAEL VIEIRA MORELLI

CONTROLE E AUTOMAÇÃO DE ESTEIRA TRANSPORTADORA

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia de São
Carlos, da Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase
em Sistemas de Energia e Automação

ORIENTADOR: Professor Doutor Dennis Brandão

São Carlos
2009

Agradecimentos

Agradeço meus pais e minha irmã, que souberam dar todo suporte para que pudesse fortalecer as minhas bases do conhecimento e assim proporcionaram o alcance de mais uma conquista.

Aos amigos da universidade, que de alguma maneira ajudaram no meu crescimento pessoal e também profissional.

A caríssima Fernanda Silva Gauy, pela ajuda, atenção e paciência oferecidas durante todas as fases do projeto e por todos os ensinamentos de vida, que sempre proporcionaram e continuarão proporcionando, que o sucesso esteja ao meu redor; a você, o meu mais sincero muito obrigado.

Ao professor Dennis Brandão, primeiramente pela amizade e ajuda prestada durante todo o trabalho, por ceder as dependências de seu laboratório e por toda paciência, não somente neste projeto, como também em todo âmbito acadêmico.

SUMÁRIO

CAPÍTULO 1
INTRODUÇÃO 1

CAPÍTULO 2
ESPECIFICAÇÕES TÉCNICAS DA ESTEIRA TRANSPORTADORA 3

CAPÍTULO 3
DESCRIÇÃO DOS PROGRAMAS UTILIZADOS NO COMPUTADOR INDUSTRIAL 10
3.1 - RSLinx 10
3.2 - Softlogix 5 11
3.3 - PLC-5 12
3.4 – PanelView 13

CAPÍTULO 4
MAPEAMENTO DAS ENTRADAS E SAÍDAS 15
4.1 - TESTE DE ENTRADAS E SAÍDAS 17

CAPÍTULO 5
CONTROLADORES LÓGICOS PROGRAMÁVEIS 20
5.1 – LINGUAGENS DE PROGRAMAÇÃO 20
5.1.1 - Linguagem de texto estruturado 21
5.1.2 - Linguagem de diagrama sequencial (SFC/Grafcet) 21
5.1.3 - Linguagem Ladder 23

CAPÍTULO 6
COMUNICAÇÃO 27

CAPÍTULO 7
RESULTADOS 30

CAPÍTULO 8
CONCLUSÃO 37

REFERÊNCIAS BIBLIOGRÁFICAS 38

ANEXOS 40

RESUMO

Nos últimos anos, visando a redução de custos e a eliminação de resíduos, muitas empresas têm investido em tecnologias de reciclagem. Nota-se que existe uma enorme preocupação com o lixo tecnológico, que são aqueles aparelhos eletroeletrônicos que estão com defeitos, ou mesmo que se encontram obsoletos no mercado.Neste trabalho, foi implementado, utilizando lógica ladder, o controle e a automação de uma esteira transportadora que integra uma célula de reciclagem de lixo tecnológico. Tal sistema é composto por um robô, um computador industrial e a própria esteira. Na última, foco deste trabalho, tem-se um motor central que está sempre em movimento, estações com pistões pneumáticos, que elevam ou rebaixam pallets e também motores que colocam os pallets para fora da parte central da esteira, enviando-os diretamente para estações de trabalho ligadas a cada uma delas. Foram verificados todos sensores, motores, parte elétrica e todo o sistema pneumático da esteira. Feito isto, iniciou-se a programação ladder no computador industrial conectado à esteira, para o controle da mesma. Em paralelo, foi estudada a forma de comunicação entre as partes envolvidas. Também foi executado o interfaceamento entre o robô da célula de reciclagem e a esteira controlada, que consiste basicamente em considerar os sensores e acionamentos do motor presente no robô, para a programação no computador industrial.

Palavras-chave: Automação, Esteira, Plc, Programação, Ladder, Reciclagem

ABSTRACT

Over the last years, many companies, aiming at reducing costs and waste disposal, have invested in recycling technologies. In fact, there's a huge concern about the electronic waste, which are defective electronic devices, or even devices that are obsolete in the market.In this work was implemented, using ladder logic, the control and automation of a transportation conveyor which integrates a cell for electronic waste recycling. This system consists of a robot, an industrial computer and the conveyor. In the conveyor, focus of this work, there is a central motor that is always in motion, stations with pneumatic pistons, which amount or demote pallets, and also motors that put the pallets out of the central part of the conveyor, sending these pallets directly to work stations related to each one of them.All sensors, motors, electrical parts and the whole pneumatic system of the conveyor were checked. After this, the ladder programming was started in the industrial computer, connected to the conveyor, as a way to control the device. In parallel, the communication between the evolved parts was being studied.Also, it was implemented the interface between the robot of the recycling cell and the controlled conveyor, which basically consists of considering the sensors and drivers of the motors connected to the robot, to use on programming the industrial computer.

Keywords: Automation, Conveyor, Plc, Programming, Ladder, Recycling

CAPÍTULO 1

INTRODUÇÃO

Nota-se, nos dias de hoje, que tem aumentado a preocupação da maior parte da população mundial quanto à questão do desenvolvimento sustentável. Um dos vilões, quando se fala em sustentabilidade, é o lixo tecnológico, que consiste de equipamentos elétricos e eletrônicos obsoletos descartados no meio ambiente, e que causam enorme impacto ambiental se não forem tratados apropriadamente.

Um dos piores casos de descarte de materiais eletro-eletrônicos é o dos telefones celulares, sendo que estima-se atualmente que o número de aparelhos ultrapassados supere 500 milhões e continue a aumentar rapidamente [1]. Estes são geralmente descartados em aterros sanitários ou incinerados, e assim liberam substâncias tóxicas (metais pesados) que se concentravam nas baterias, circuitos impressos, displays de cristal líquido, nas carcaças de plástico ou até mesmo na própria fiação [2].

Por se tratar de produtos químicos de alto risco para a saúde humana, estes resíduos devem ser devidamente tratados. Porém, este processo é de alta complexidade e demanda o uso de tecnologias avançadas, devido à diversidade de materiais de sua composição e à periculosidade das substâncias tóxicas [17].

Atualmente, até existem nos países desenvolvidos campanhas para arrecadação de aparelhos celulares obsoletos para reciclagem, o que é uma prática extremamente necessária, visto que a média de tempo de permanência com os telefones para as pessoas no mundo é de apenas 20 meses [2]. Depois deste período, estes aparelhos geralmente são descartados no meio ambiente, sem nenhum tratamento, levando a sérios danos ambientais.

Frente a esta realidade, mostra-se o objetivo principal deste trabalho: colocar em funcionamento pleno a esteira e criar um programa em linguagem Ladder para automatizar a esteira. Para isso será utilizada uma célula de reciclagem automatizada, que consiste de um computador industrial, onde se concentra a lógica do sistema, uma esteira transportadora e um robô que realiza a separação das peças do celular.

Para alcançar tal objetivo, num primeiro momento, foi dada prioridade para o funcionamento do hardware, ou seja, foi colocado em funcionamento toda a parte elétrica e pneumática da esteira, todos os motores, válvulas e sensores da

célula(incluindo os presentes no robô). Finalizada esta parte, teve início o trabalho com o software no computador industrial, onde foi estudado o modo de comunicação do mesmo com a célula de reciclagem e posteriormente foi criado um programa, utilizando lógica ladder, para automatizar o referido equipamento.

É importante ressaltar que esta unidade de reciclagem foi doada pela *Technische Universität Berlin* (Universidade Técnica de Berlim), pois a mesma tem uma parceria com a Universidade de São Paulo (Campus de São Carlos), através do Instituto Fábrica do Milênio. Esta parceria tem por objetivo a conclusão de diversos projetos na área do gerenciamento do ciclo de vida de equipamentos elétricos e eletrônicos. Também fazem parte deste acordo o Núcleo de Manufatura Avançada (NUMA) da USP-São Carlos, o Centro de Caracterizaçao e Desenvolvimento de Materiais (CCDM) da Universidade Federal de São Carlos e a Universidade Federal de Rio Grande do Norte.

CAPÍTULO 2

ESPECIFICAÇÕES TÉCNICAS DA ESTEIRA TRANSPORTADORA

Como foi dito anteriormente, a célula de reciclagem é composta por um robô, a esteira transportadora e um computador industrial. Com isso, nota-se que é importante dar uma visão geral sobre a célula de reciclagem e isto fica bem claro na figura 2.1:

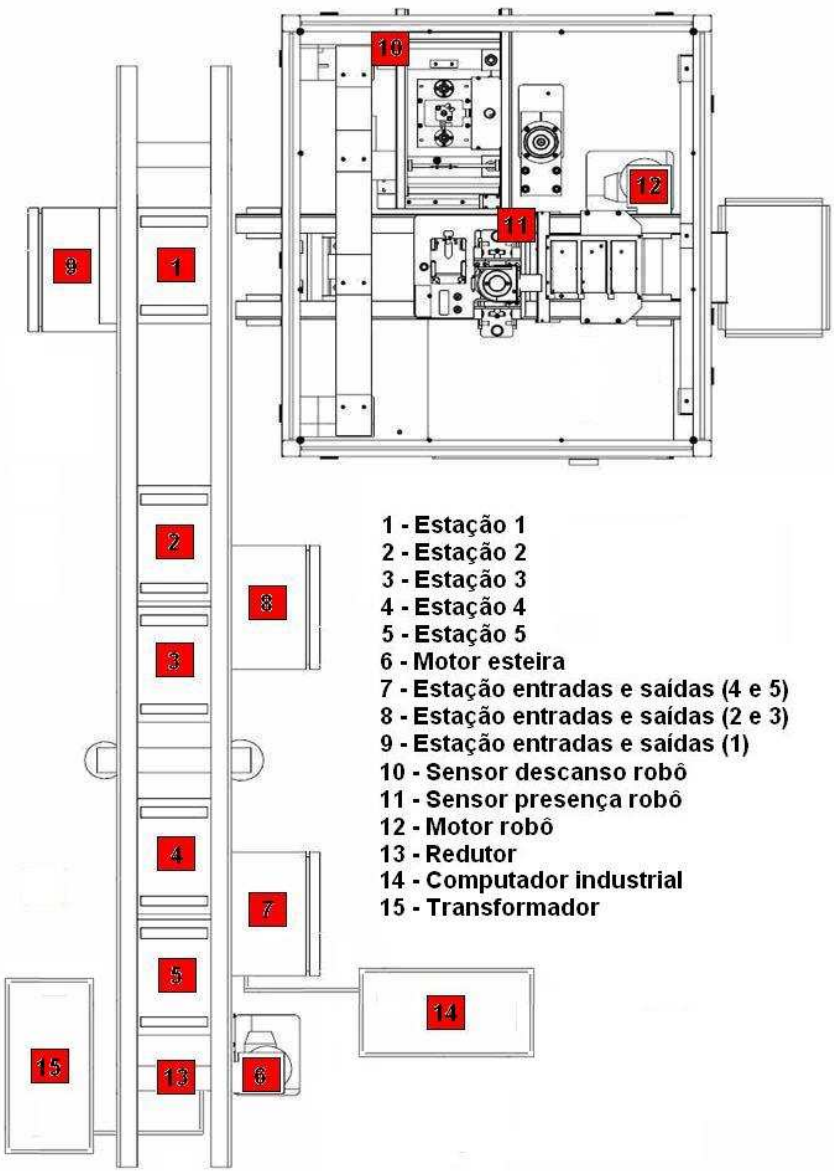


Figura 2.1 – Visão geral dos componentes da célula de reciclagem

Faz-se necessário uma explanação sobre o funcionamento geral da célula de reciclagem a partir deste momento, e um bom entendimento se dá com as figuras de 2.1 até 2.14 e com a própria explicação a seguir.

A unidade é composta de um robô (figura 2.3), um computador industrial (figura 2.4) e por uma esteira transportadora (figura 2.2), que executam respectivamente as seguintes atividades: desmontagem do aparelho celular, a interação lógica do sistema e por último o transporte de pallets.

Figura 2.2 - Esteira

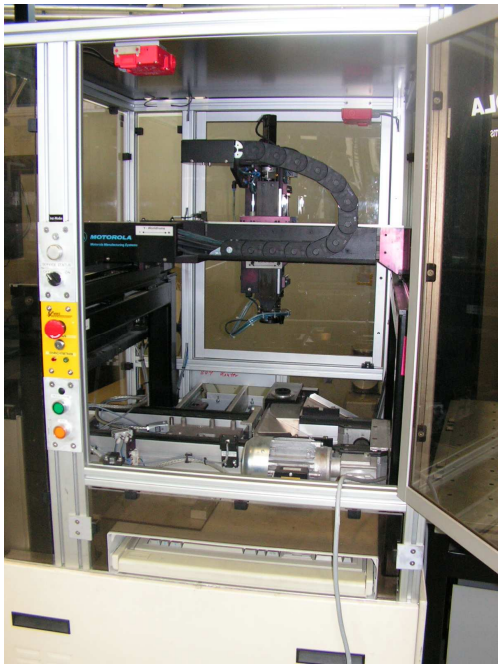


Figura 2.3 - Robô

Figura 2.4 – Computador Industrial

A esteira é alimentada eletricamente pelo transformador da figura 2.5 e temos um motor central (figura 2.6), que funciona sempre e faz com que as correias girem no sentido da estação 5 para 1, sendo este o rumo de entrada que os pallets devem seguir para entrar no robô.

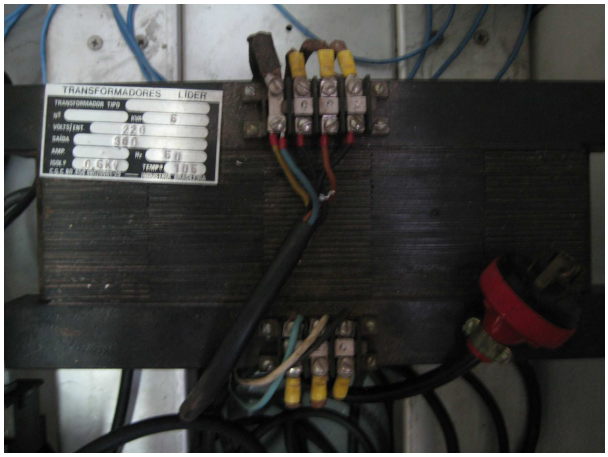


Figura 2.5 - Transformador

Figura 2.6 – Motor central da esteira

Em cada estação existem caixas que monitoram as entradas e saídas do sistema (figura 2.9), sendo que elas verificam o estado dos sensores de presença (figura 2.10) e dos sensores de movimentação (figura 2.11) dos pistões presentes no sistema pneumático (figura 2.14), que mostram se os pistões elevaram ou rebaixaram a estação, segurando assim os pallets na determinada estação. Ou seja, quando se deseja manter o pallet na estação aciona-se o pistão e com isso o sensor de movimentação é acionado(quando a estação sobe). De acordo com a presença de pallets nas estações, podem ser acionados os motores presentes em cada uma delas (figura 2.8) e estes giram correias menores no sentido perpendicular ao do eixo principal da esteira, levando os pallets para fora da esteira ou trazendo-os de volta para o eixo principal. Na estação 1 da figura 2.1, existe um motor (como o da figura 2.8) que é acionado juntamente com o motor do robô (figura 2.7) para que o pallet entre no robô para confeccionar a atividade de desmontagem, quando o sensor de presença do robô (figura 2.12) é acionado.

Figura 2.7 – Motor do robô

Figura 2.8 – Motor da estação 5

Figura 2.9 – Estação de entradas e saídas

Figura 2.10 – Sensor de presença do pallet na esteira

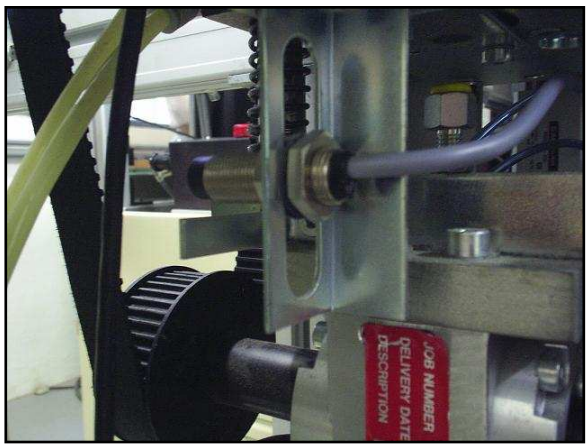


Figura 2.11 – Sensor de movimentação da estação

Assim que o robô encerra a desmontagem, ele retorna para uma posição de descanso e ativa um sensor (figura 2.13), que faz com que o pallet seja devolvido para estação 1 com o acionamento dos motores em sentido reverso.

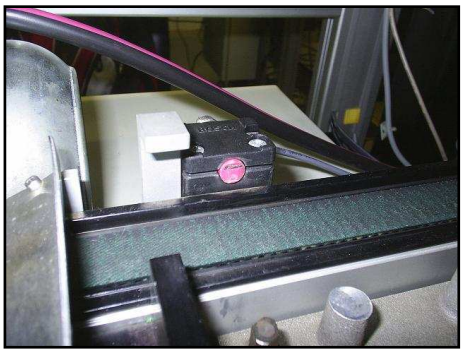


Figura 2.12 – Sensor de presença do robô

Figura 2.13 – Sensor de descanso do robô

Figura 2.14 – Sistema pneumático da esteira

Então, de uma maneira geral, o que esperava-se na confecção deste projeto, era automatizar todos esses elementos, fazendo com que os pallets fossem da estação 5 até a estação 2, aguardassem um sinal de liberação do robô, para enviar um pallet para estação 1 e posteriormente para dentro do robô. Depois de encerrada a

atividade no interior do robô, motores das estações enviariam o pallet de volta para a parte central da esteira, que o faria seguir adiante e assim forma-se um ciclo automatizado de entrada e saída de pallets para reciclagem dos aparelhos celulares, diminuindo desta maneira a ação repetitiva do ser humano.

CAPÍTULO 3

DESCRIÇÃO DOS PROGRAMAS UTILIZADOS NO COMPUTADOR INDUSTRIAL

Basicamente foram utilizados 4 programas no computador industrial para conclusão do projeto: RSLinx(comunicação entre os softwares), Softlogix(simulador de PLC), PLC-5(criação e edição de programas Ladder) e Panelview(controle e monitoramento da esteira – IHM). Um detalhamento sobre cada um deles será dado a seguir.

3.1 - RSLinx

Este programa tem por finalidade realizar a comunicação entre os mais diversos componentes e softwares da *Allen Bradley*, funcionando basicamente como uma biblioteca de drivers que realiza tal ligação.

O programa prevê acesso a uma grande variedade de aplicativos da Allen Bradley, como por exemplo RSLogix, RSNetWorx e RSView32 para o caso de IHM´s(Interfaces Homem-máquina). Também pode-se utilizar na aquisição de dados por parte de alguns programas como Microsoft Excel, Microsoft Acess e Visual Basic [3].

O RSLinx apresenta uma interface gráfica muito amigável para o usuário, possibilitando navegar através das redes de comunicação dos programas e dispositivos existentes no programa [3]. É facilmente configurado para os dispositivos do projeto em questão, sendo que torna-se possível até mesmo visualizar códigos em linguagem ladder feitas em outro aplicativo.

É fácil notar pela figura 3.1 que, basta colocar os hardwares e softwares que precisam se comunicar na mesma janela e assim obtém-se uma ligação entre os mesmos [4].

Figura 3.1 – Tela principal do programa RSLinx

Vale ressaltar em quais condições o RSLinx foi utilizado para o bom desempenho do sistema como um todo. Como drivers de comunicação e configuração utilizou-se: Softlogix5 (se for modificado um código ladder do PLC5 no computador industrial, remover o driver DH+ pois senão ocorrerá um conflito.

3.2 - Softlogix 5

O Softlogix possibilita o computador industrial reconhecer e colocar em funcionamento qualquer lógica elaborada em outro programa em linguagem ladder, diagrama funcional de blocos, texto estruturado e SFC (sequential function chart) [5].

O Softogix 5, instalado no computador industrial, possui o driver KTXD para que o cartão 1784-KTXD presente no mesmo, possa ser configurado com a função de controlar entradas e saídas, sendo portanto utilizado neste projeto para controle remoto das E/S. Além disso, pode ser configurado como DH+(data highway plus) através do programa do RSLinx, tendo assim a função de se comunicar mais rapidamente com um PLC(programmable logic controller). O software também possibilita que as portas seriais do computador industrial sejam configuradas para simular as portas serias de um PLC [5].

A tela principal de monitoramento do Softlogix pode ser vista na figura 3.2 [5]:

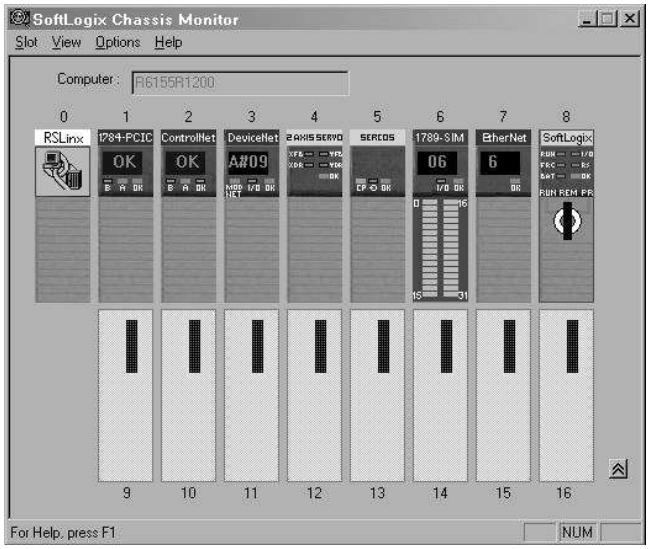


Figura 3.2 – Tela principal do programa Softlogix

O aplicativo é como um controlador que faz parte de um ambiente Logix (ambiente de lógica), sendo que suporta todos os programas Allen Bradley relacionados à lógicas de automação e também algumas instruções diretas de movimento. Esta versão do programa suporta os seguintes componentes [5]:

- Software de programação RSLogix e PLC5, sendo que podem ser produzidos(on-line e off-line) lógicas em ladder, diagrama funcional de blocos, texto estruturado e SFC.
- 1784-KTXD, 1784-PCICS e 1784-PM02AE, que possibilitam a comunicação e consequentemente controle de entradas e saídas.
- Tem disponibilidade para acesso à uma porta Ethernet que faça controle de entradas e saídas sobre uma rede Ethernet/IP.

Portanto, conclui-se que este programa é o componente principal dentro do computador industrial, visto que ele funciona como um simulador de um controlador lógico programável.

3.3 - PLC-5

Este é o programa presente no computador industrial que permite criação e edição de programas lógicos, para o controle da esteira e do robô, utilizando diversas linguagens de programação. É possível programar em SFC(sequential function chart), texto estruturado e também em Ladder, sendo que esta última foi a única utilizada neste trabalho [6].

Apesar do Softlogix ser o software essencialmente necessário para o funcionamento do projeto, o PLC-5 é onde se produz a lógica de automação da esteira e do robô. Portanto, a maior parte do tempo neste trabalho foi gasta produzindo tal lógica no PLC-5.

A figura 3.3 mostra a tela principal do programa [6]. Pela imagem, nota-se na segunda linha do cabeçalho o nome do programa que está carregado no momento(lado esquerdo). A terceira linha do cabeçalho mostra o dispositivo de comunicação atual, seguido pelo modo de endereçamento do processador.

No menu existente na figura 3.3, pode-se selecionar a opção desejada: acessando o item "*F1) Select Program/PLC-5 Addr*" pode-se selecionar um programa já elaborado ou também criar um novo programa. Feito isto, pode-se trabalhar em modo offline usando a opção *"F2) Offline Programming/Doc*" e em modo online selecionando *"F3) Online Programming/Doc*". Outro ítem importante é o "*F6) Up/Download Program to PLC-5*" pois é nele que consegue-se colocar o programa para rodar ou mesmo copiar o programa que está armazenado na esteira em um dado momento.

Figura 3.3 – Tela principal do programa PLC-5

Mais detalhes sobre a linguagem utilizada e de como foi elaborada a programação, serão dados no capítulo 5.

3.4 – PanelView

O software responsável pelo monitoramento e controle da esteira é aberto automaticamente ao inicializar-se o computador industrial e é conhecido por PanelView. É o programa original do fabricante da esteira (Motorola) e na tela

principal, evidenciada na figura 3.4, temos uma enorme quantidade de opções para configuração, status, manutenção, ajuda, controle, versão do software e inclusive pode-se ter informações sobre situações de emergência, como por exemplo uma falha em um dos sensores [6].

Para acessar boa parte dos itens nesta tela principal, que está destacada na figura 3.4, é necessário utilizar uma senha, que no caso é 2008. Pode-se fechar o programa através de um botão invisível(que é desta maneira por segurança) ou mesmo pressionando as teclas “ctrl+alt+delete” e encerrando o programa, o que leva o usuário diretamente para o ambiente do *Microsoft Windows*, onde podem ser usados os outros programas descritos anteriormente.

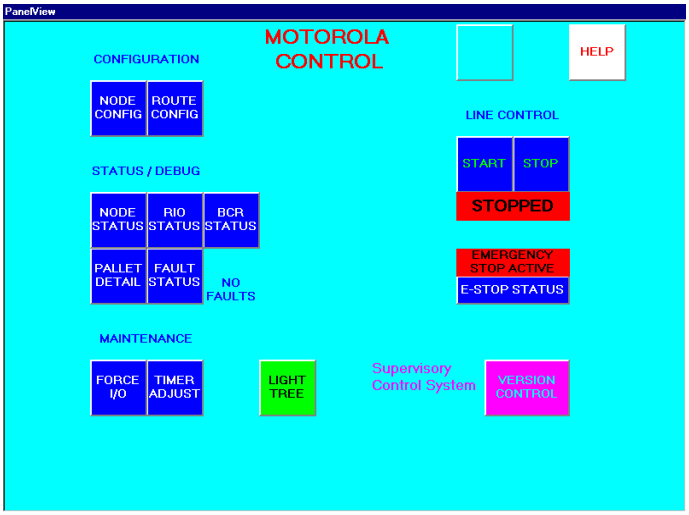


Figura 3.4 – Tela principal do programa PanelView

Vale ressaltar que, de certa forma, o desconhecimento de tal senha atrasou bastante o andamento do projeto, pois não era possível ter acesso aos outros programas do computador industrial. Outro fator que determinou ocasionou lentidão no projeto é o fato de que o computador industrial é touch screen, funcionando praticamente como uma IHM (interface homem-máquina), e sendo assim não tem-se a mesma rapidez do que com a utilização de um mouse.

CAPÍTULO 4

MAPEAMENTO DAS ENTRADAS E SAÍDAS

Primeiramente, é necessário fazer alusão ao programa PLC-5 explicado anteriormente no capítulo 3, utilizado para confeccionar a lógica ladder. Portanto, abaixo (figura 4.1) tem-se uma imagem da tela de programação do PLC-5, em modo OFFLINE, como pode ser visto no canto esquerdo superior da tela [6]. Evidenciam-se 2 detalhes que neste momento tem como informações importantes apenas o seguinte :

- **I:033_12**
- **O:032_01**

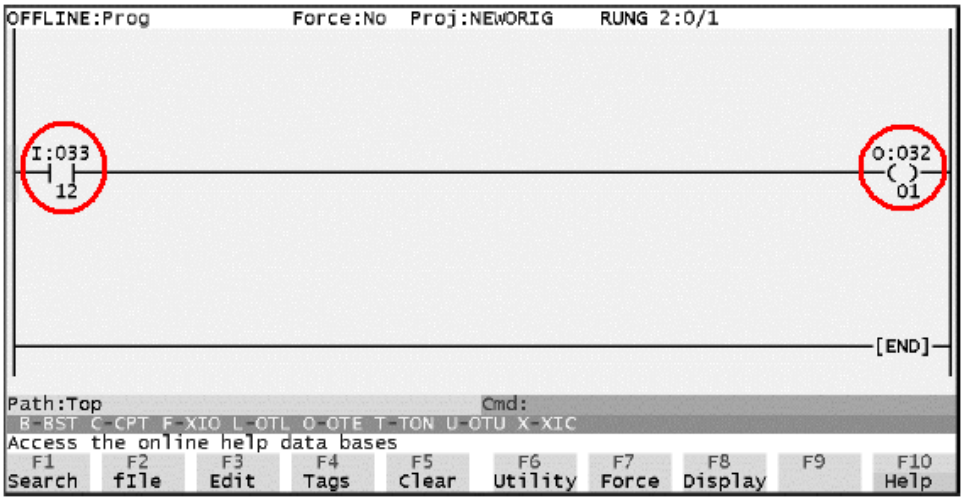


Figura 4.1 – Tela de programação do PLC-5 em modo offline

Uma noção melhor sobre como funciona a programação ladder será dada no capítulo 5. Porém, a partir deste momento é preciso dizer o que cada um destes detalhes evidenciados na figura 4.1 representam, para um bom entendimento do funcionamento do sistema. Nos detalhes, notamos que o elemento da esquerda é uma entrada (conhecida como contato), podendo ser um sensor e o elemento da direita é uma saída(conhecida como bobina), podendo ser por exemplo um motor que é acionado através de um contator. Logo, na programação efetuada, foi criada uma lógica de interação entre os elementos de entrada e saída da esteira transportadora e do robô, para que a célula de reciclagem funcionasse do modo desejado.

Alguns componentes específicos deste projeto, que atuam como entrada ou saída foram diretamente utilizados na programação estão descritos abaixo:

- Sensor de presença do pallet no robô

- Sensor de posição final(sensor de descanso do robô)
- Motores rodando em ambas direções
- Avançar/recuar pistão das estações
- Sensor de movimentação da estação
- Sensor de presença do pallet na esteira

Pela tela principal de programação, tem-se que o sistema de mapeamento e endereçamento baseiam-se na seguinte idéia: I:RRG_BB ou O:RRG_BB, sendo que as letras maiúsculas representam:

- I = ENTRADA
- O = SAÍDA
- RR = NÚMERO DO RACK
- G = NÚMERO DO GRUPO
- BB = BIT DE CONTROLE

Em cada uma das estações de entradas e saídas estão presentes componentes da Allen Bradley conhecidos como Flex I/O, responsáveis pelas entradas e saídas, como o próprio nome já diz. Dois modelos foram utilizados especificamente: Flex I/O 1794 IB-15 para entradas e 1794 OV-15 para saídas, sendo que para uma estação tem-se um componente de cada.

Logo, no total temos 5 Flex I/O do primeiro modelo e 5 Flex I/O do segundo alocados na esteira. Vale lembrar que os sensores e motores do robô foram conectados na Flex I/O da estação 1.

Utilizando o software PLC-5, obteve-se os seguintes endereços (tabela 4.1) para as diversas entradas e saídas da célula de reciclagem:

Tabela 4.1 – Endereços correspondentes das entradas e saídas no robô e na esteira

Robô	
Sensor presença pallet	I:050_04
Sensor posição final	I:050_00
Motor direto	O:051_02
Motor reverso	O:051_01

Estação 1	
Sensor de movimentação	I:050_01
Sensor presença pallet	I:050_03
Recuar pistão	O:051_04
Motor direto	O:051_02
Motor reverso	O:051_01
Avançar pistão	O:051_03

Estação 2	
Sensor de movimentação	I:042_01
Sensor presença pallet	I:042_03
Recuar pistão	O:043_04
Motor direto	O:043_01
Motor reverso	O:043_02
Avançar pistão	O:043_03

Estação 3	
Sensor de movimentação	I:044_01
Sensor presença pallet	I:044_03
Recuar pistão	O:045_04
Motor direto	O:045_02
Motor reverso	O:045_01
Avançar pistão	O:045_03

Estação 4	
Sensor de movimentação	I:046_01
Sensor presença pallet	I:046_03
Recuar pistão	O:047_04
Motor direto	O:047_02
Motor reverso	O:047_01
Avançar pistão	O:047_03

Estação 5	
Sensor de movimentação	I:040_01
Sensor presença pallet	I:040_03
Recuar pistão	O:041_04
Motor direto	O:041_01
Motor reverso	O:041_02
Avançar pistão	O:041_03

4.1 - TESTE DE ENTRADAS E SAÍDAS

Todas as entradas e saídas foram testadas exaustivamente de 2 maneiras. Na primeira delas, deve-se forçar os dispositivos, observar o que ocorre no software instalado no computador industrial e se os respectivos leds se acendiam na esteira transportadora.

Para isso, utilizou-se o modo de programação ONLINE, acessado pressionando a tecla F3, no menu central da figura 3.3. Fazendo isso, o programador é direcionado para a tela da figura 4.2 e neste local, nota-se que o PLC-5 mostra o estado(ativas ou não) das entradas e saídas do programa em vigor [6].

Sabendo que existe a possibilidade de monitorar o estado das E/S, foram efetuados alguns testes manuais, que de maneira simples puderam determinar se os dispositivos estavam em correto funcionamento.

Nas entradas, que no caso são os sensores, colocou-se próximo dos mesmos algum material metálico e observou-se que cada um deles respondia da seguinte maneira: um led interno do dispositivo se acendia, nas estações de entradas e saídas os leds da Flex I/O se acendiam e também no computador industrial era possível observar que o dispositivo estava ativo.

As saídas(válvulas, contatores e motores) eram ativadas de acordo com a lógica de programação criada. As válvulas acendiam um led interno, acionavam o seu respectivo pistão que elevava a estação e era possível notar que estavam ativas na tela do computador industrial. Com os motores, notou-se que os mesmo giravam quando eram acionados pela lógica e que estavam ativos como saída na tela do computador industrial. Tanto para as válvula quanto para os motores, notou-se que nas estações de entradas e saídas, seu respectivo led se acendia no dispositivo Flex I/O.

Efetuando este teste, verificou-se que todas as entradas e saídas presentes na célula de reciclagem, ou seja, sensores, motores e válvulas, estão funcionando corretamente.

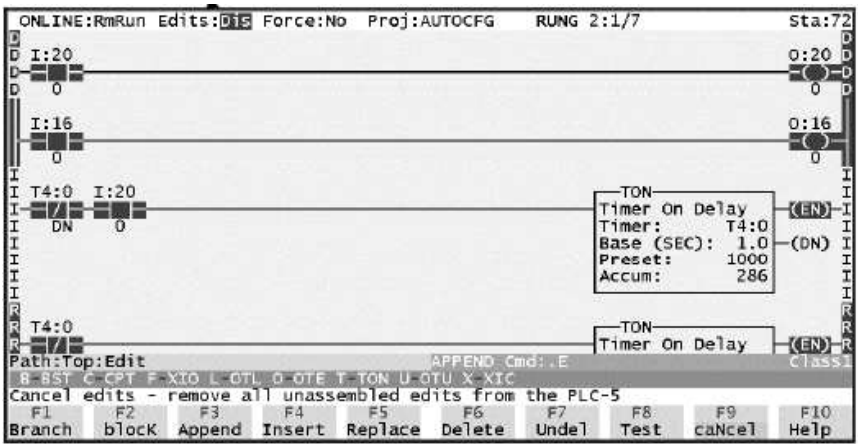


Figura 4.2 – Tela de programação online do PLC-5

Outra maneira de efetuar o teste é fazendo justamente o contrário, ou seja, forçar as entradas e saídas diretamente no computador industrial, observar o que acontece com os dispositivos e verificar se o led de cada um deles se acende nas caixas Flex I/O.

Isto pode ser feito acessando o modo de programação OFFLINE, a partir da tecla F2 do menu principal do programa PLC-5 [6]. Ao entrar nesta tela, tem-se na

parte inferior da tela a opção F7 – FORCE, como pode ser visto na figura 4.1. Pressinando tal tecla, o usuário é direcionado outra tela (figura 4.3), onde pode setar todas as entradas e saídas em ativas ou inativas. Para mudar de entradas para saídas, o usuário deve pressionar F2 - OUTPUTS e quando estiver no menu com as saídas deve pressionar F1 – INPUTS para retornar às entradas.

Para deixá-las ativas, basta escrever 1 no lugar do ponto(.) que existe no endereço específico de cada dispositivo e para desativá-las basta escrever 0 no mesmo local. Para que as mudanças tenham efeito, é necessário pressionar a tecla F6 (ENABLE) e caso queira voltar ao modo normal basta apertar F7 (DISABLE).

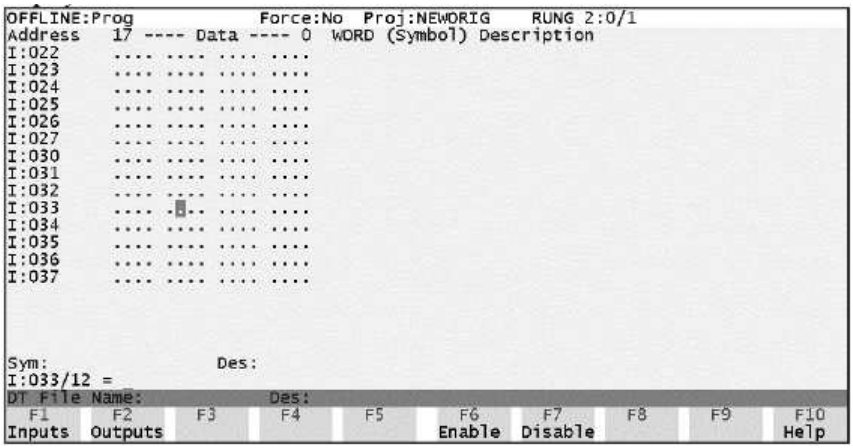


Figura 4.3 – Tela de testes do PLC-5

Utilizando-se desta ferramenta, todos os sensores, motores e válvulas foram testados e responderam corretamente de acordo com o que era esperado de cada um deles. Além dos dispositivos que realmente existem e estão interligados fisicamente à esteira, é possível testar também todas as entradas e saídas que estão livres na caixa Flex I/O e para isso basta forçá-las em 1 no programa PLC-5 e verificar se o seu próprio led acende na estação de entradas e saídas.

CAPÍTULO 5

CONTROLADORES LÓGICOS PROGRAMÁVEIS

Deve-se lembrar que o trabalho de automação industrial era antigamente realizado por meio de circuitos e relés acionadores. A partir de desenhos representativos dos controles industriais, eram elaboradas listas de trabalho e posteriormente eram montados em armários elétricos todos esses comandos. Cada comando , simbolizado através de um circuito elétrico, era traduzido em uma lista de instruções e desta para uma montagem, alojada normalmente em painéis elétricos. Assim, quando era necessário efetuar uma modificação no comando, por qualquer motivo, isso implicava em um rearranjo na montagem, geralmente cansativo, trabalhoso, demorado e dispendioso. Muitas vezes, uma modificação implicava em uma reforma total nos painéis elétricos. As substituições das peças em caso de manutenção de montagem, tornavam-se difíceis, ou até mesmo impossível [8].

Os controladores lógicos programáveis, assim como também o simulador empregado neste projeto, vieram trazer um grande avanço tecnológico, facilidade para as manutenções, redução de tempo de engenharia e muitos outros benefícios. Os controles de processos industriais ou automação da manufatura é sem dúvida, uma das aplicações de maior impacto. O controlador programável pode ao mesmo tempo automatizar uma grande quantidade de informações, substituindo assim o homem com mais precisão, confiabilidade, custo e rapidez [9].

5.1 – LINGUAGENS DE PROGRAMAÇÃO

Existem atualmente diversas linguagens de programação que podem ser usadas em controladores programáveis. Todas elas devem ser padronizadas pelo orgão IEC (*International Electrotechnical Committee*), e mais especificamente para este caso tem-se a norma IEC 1131-3 – *Programming Languages* que dita quais regras devem ser seguidas [10].

Segundo esta norma, as linguagens de programação podem ser divididas em:

Tabela 5.1 – Linguagens de programação em automação industrial

Classes	Linguagens
Tabulares	Tabela de decisão
Textuais	Lista de instrução (Instruction List)
	Texto estruturado (Strutured Text)
Gráficas	Diagrama de blocos funcional
	Ladder (Diagrama de relés)
	SFC (Sequential function chart)

De todas linguagens descritas na tabela 5.1 [11], foi visto no capítulo 3, que é possível, através do PLC-5, programar em SFC, texto estruturado e também em Ladder, sendo que esta última foi utilizada neste projeto. Apesar de apenas uma ter sido utilizada, uma explanação será dada sobre cada uma delas a seguir.

Foi realizado um estudo para evidenciar qual seria a melhor linguagem a ser utilizada para programação, procurando dentre as opções existentes no computador industrial a mais adequada ao nível de complexidade (relativamente baixo) do projeto.

5.1.1 - Linguagem de texto estruturado

A mais rudimentar das três anteriores é o texto estruturado, e nota-se que é uma linguagem de alto nível no formato de texto em que não existe ordem de execução. É basicamente como Pascal ou Basic, linguagens nas quais se atribui novos valores às variáveis do lado esquerdo das instruções [11].

Um exemplo simples, que nos mostra diretamente o funcionamento de tal linguagem é: 05 = I1 AND NOT I2 AND I3 OR I4

Nota-se que não é uma linguagem muito gráfica, portanto, seria difícil utilizar a mesma para a programação da esteira.

5.1.2 - Linguagem de diagrama sequencial (SFC/Grafcet)

Sequential function chart, SFC ou Grafcet são os diversos nomes para uma mesma linguagem de programação complexa, de alto nível, muito usada para controle de sistemas. Um programa em SFC oferece já uma visão geral sobre o sistema de controle, onde cada bloco básico(passos) são programas inteiros geralmente construídos em ladder [11]. A utilização desta linguagem possibilita que trabalhos

complicados de programação sejam reduzidos a trabalhos menores e muito mais gerenciáveis.

O programa utilizando sequential function chart é feito em passos, ações, transições e arcos. No figura 5.1, tem-se que P1, P2 e P3 são os passos, ou seja, são atividades que serão executadas até que as transições possibilitem mudar de um passo para outro. T1,T2 e T3 são as transições, que são melhor entendidas como condições para que ocorra a mudança de um passo a outro. Temos que A1 é uma ação que deverá ser executada no decorrer do passo 2 e os arcos são as linhas que estão ligando os diversos elementos [11].

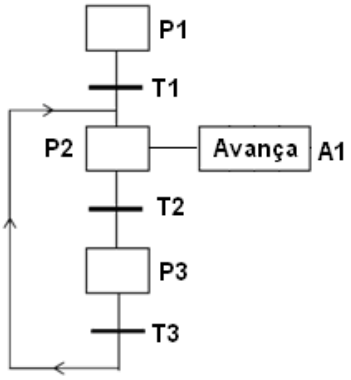


Figura 5.1 – Exemplo de programa em SFC

O programa em SFC apresenta diversos benefícios, dentre eles [12]:

- Permite uma visualização rápida e mais objetiva do desenvolvimento e operação da automação em questão.
- A método de organização da programação facilita a sua leitura e seu entendimento por parte de outras pessoas.
- Modularidade
- Mais facilidade para encontrar problemas, devido a natureza modular dos programas
- Geralmente consomem muito menos memória para sua execução por utilizar funções modularizadas e tem-se diminuição no tempo de varredura, pois os passos não utilizados não são carregados na memória.

Para ilustração, tem-se disponível a figura 5.2 [6], que mostra como é feito um programa no PLC-5 utilizando tal linguagem.

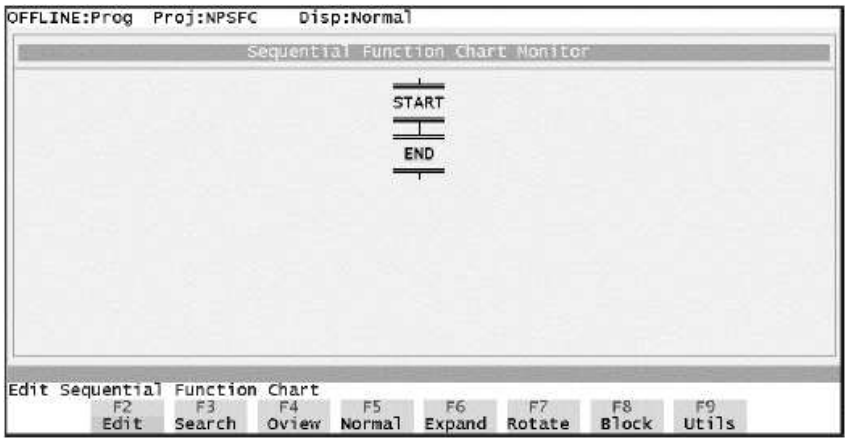


Figura 5.2 – Exemplo de programa em SFC criado no PLC-5

Apesar dos diversos benefícios de programar em SFC, o programador precisa ter muita experiência profissional para utilizar tal linguagem, pois a mesma inclui também conhecimento prévio de outras linguagens usadas em automação, principalmente ladder. Por estes motivos, a utilização desta linguagem tornou-se inviável.

5.1.3 - Linguagem Ladder

Também conhecida como linguagem de diagrama de contatos, a linguagem ladder é a mais amplamente utilizada quando de trata de automação industrial e permite programar desde funções binária simples até as mais complexas atividades matemáticas. Seu princípio de funcionamento é o mesmo dos diagramas de relés e contatos, e nota-se que independente da complexidade do programa de aplicação, há certos fundamentos da linguagem que são imprescindíveis para um desenvolvimento adequado e que são válidos genericamente a todos os controladores lógicos programáveis [12].

A linguagem consiste de um diagrama lógico em que tem-se basicamente 2 componentes: contatos e bobinas. Os contatos representam as entradas, que podem ser sensores mecânicos, indutivos, capacitivos, térmicos, chaves fim-de-curso e até mesmo taco-geradores, enquanto que as bobinas representam as saídas, que podem ser válvulas, lâmpadas, contatores e relés convencionais.

A figura 5.3 representa os 2 elementos mais utilizados e um terceiro que é apenas uma variação do contato normalmente aberto.

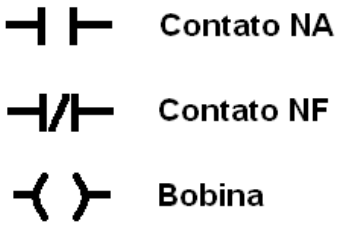


Figura 5.3 – Elementos básicos da programação ladder

Segundo [11], existem regras básicas para constituição do diagrama ladder (ou diagrama de contatos):

- Os contatos estão sempre à esquerda e as bobinas mantém-se sempre à direita, próximos às linhas verticais, conhecidas como linhas-mãe.
- Todos os elementos ficam contidos entra duas linhas verticais e tem-se uma sequência semelhante aos degraus de uma escada.
- A sequência de causa e efeito, além de estarem de acordo com a lógica, vão da esquerda pra direita e de cima para baixo.
- As bobinas só são acionadas se a linha horizontal correspondente estiver ativa. Logo, uma bobina é acionada de acordo com a lógica dos contatos à esquerda de sua linha.

Nota-se que, para que uma bobina (ou qualquer outro elemento de saída) seja acionada, é necessário que ela tenha diferença de potencial. Logo, emprega-se o conceito de corrente lógica fictícia, ou seja, supondo que entre as barras verticais exista uma diferença de potencial (a barra da esquerda com potencial maior que a barra da direita), haverá a circulação de corrente da esquerda para a direita se a lógica de controle permitir tal fato [13].

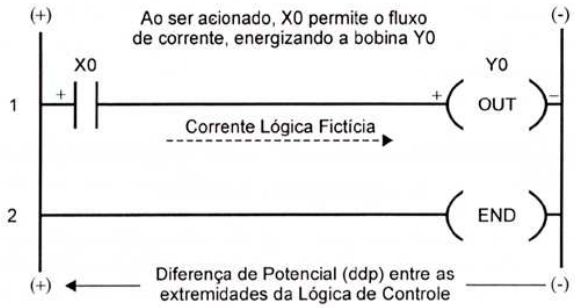


Figura 5.4 – Exemplo de programa evidenciando sentido da corrente fictícia

Com a figura 5.4, percebe-se que se o Contato de XO (NA), que pode ser um botão, estiver acionado, a bobina de Y0 será energizada, atuando esta saída. Caso contrário, a bobina de YO não será energizada pois não tem-se uma ddp, mantendo

então a bobina desligada. É importante notar que o sentido da corrente lógica fictícia é sempre, e tão somente, da esquerda para a direita, não existindo a possibilidade de fluxo em sentido contrário.

Vale lembrar que cada elemento básico(contato ou bobina, por exemplo) da lógica de controle representa uma instrução da linguagem ladder sendo alocada em um endereço específico e consumindo uma quantidade determinada de memória (word) disponível para armazenamento do programa de aplicação, conforme a CPU utilizada [12]. Um mesmo símbolo gráfico da linguagem ladder (contato normalmente fechado, por exemplo) pode representar instruções diferentes, dependendo da localização na lógica de controle.

Além dos elementos básicos, que constituem praticamente uma lógica binária para a programação, existem também elementos como por exemplo: temporizadores, contadores, blocos de função, podendo até mesmo realizar operações lógicas e algébricas.

É importante dar uma visão geral de como criar um programa em linguagem ladder no software utilizado para tal fim, pois esta foi a linguagem escolhida. Para isso, basta entrar na primeira opção da figura 3.3 e selecionar o ítem para criar um novo programa. Depois de criado o programa, o usuário deve selecionar o modo de programação OFFLINE. A partir deste ponto, o PLC-5 abrirá a janela da figura 5.5, e então cabe ao programador selecionar o opção F1) Ladder File, para programar em ladder [6]:

Figura 5.5 – Menu para escolha da linguagem de programação

Selecionando a opção F1, o programador é direcionado para a tela seguinte, na qual deve-se pressionar F4 – INSERT para inserir os elementos de entrada e saida. É neste local que o usuário consegue adicionar contatos, bobinas e outros itens que são necessários para a programação, conforme pode ser visto na figura 5.6. Neste ponto, já deve-se saber o endereço de cada componente para que a lógica seja corretamente implementada, sendo que os endereços que foram usados neste projeto estão bem determinados no capítulo 4 – mapeamento de entradas e saídas.

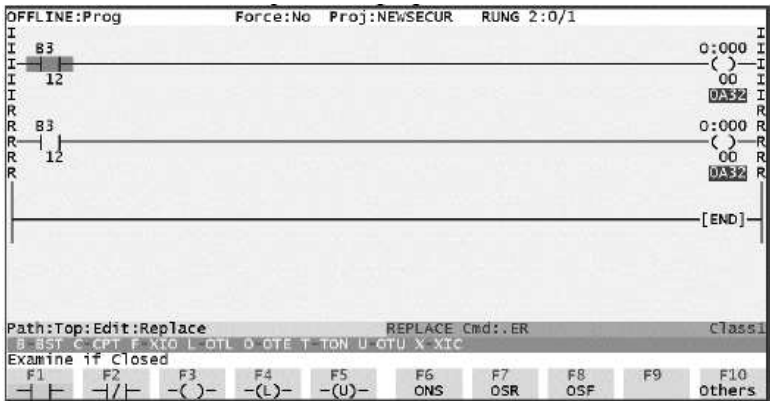


Figura 5.6 – Tela de programação offline do PLC-5, com elementos básicos na parte inferior

Tomaram-se como base alguns programas já existentes no computador industrial para compreender melhor o funcionamento do PLC-5 e foi possível assim criar o programa que automatiza a esteira. Para evitar a lentidão de programar diretamente no computador industrial, o programa foi feito utilizando um computador pessoal através do software CX-Programmer, da OMRON.

O fato de utilizar tal software, em um computador muito mais rápido, agilizou muito a programação, pois podia-se realizar testes forçando entradas e saídas no próprio computador, sem haver necessidade do software estar conectado ao hardware da esteira. Depois que esta etapa ficou pronta, foi necessário apenas criar o programa no PLC-5, adaptando-o de maneira simples, para que a lógica funcionasse na esteira transportadora.

CAPÍTULO 6

COMUNICAÇÃO

Para que fosse possível a comunicação do computador industrial com a esteira transportadora e o robô, foi utilizada a placa de comunicação 1784-KTXD do fabricante Allen-Bradley. Tal placa realiza a interface de comunicação, possibilitando a transmissão de dados, assim como também gerencia redes locais e diagnostica falhas na comunicação.

Este dispositivo deve ser conectado em um slot de 16 bits ISA ou EISA, e tem-se uma visão superior da placa na figura 6.1, mostrando como detalhe um jumper que deve ser colocado na posição 2-3, para que a placa funcione no modo de operação de 16 bits. A posição 1-2 do jumper, faz com que a placa trabalhe em modo 8 bits, que somente é usado quando se tem problemas na utilização em 16 bits [14].

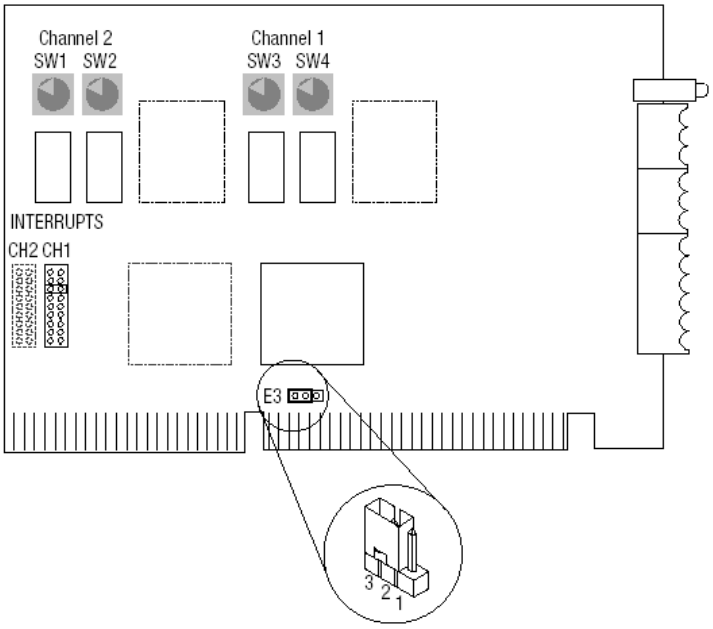


Figura 6.1 – Visão superior da placa 1784-KTXD, com jumper no detalhe

É necessário também explicitar um outro detalhe na figura 6.1, que nos mostra as chaves SW3 e SW4(para o canal 1). De acordo com a tabela 6.1, retirada de [14], o endereço que deve ser utilizado é o D700. Portanto, a chave SW3 deve estar na posição D e a chave SW4 deve estar na posição 7, quando estamos utilizando tal placa para comunicação através do canal 1.

Tabela 6.1 – Endereços para utilizar nas chaves das placas de comunicação

Catalog Number	Channel	Address
1784-KTS	1	D700:
1784-KTX	1	D700:
1784-KTXD	1	D700:
	2	D600:

A 1784-KTXD possui dois canais de comunicação, que funcionam em dois modos: DH+(Data Highway Plus) e DH-485(Data Highway baseado no padrão RS-485) [14]. O modo de trabalho depende da maneira que se conectam os cabos à placa, sendo que um detalhamento a este respeito pode ser observado na vista frontal que está na figura 6.2.

Nota-se através de [14], que este hardware tem compatibilidade com os seguintes softwares da Allen-Bradley: 6200 Series, 1784-KTx Scanner Reference Set, Interchange, AI-PLC-5, Softlogix, RSLogix 5 e RSLogix 500 via RSLinx. Fica claro no manual que os softwares de interface Allen-Bradley são responsáveis por gerenciar a recepção e transmissão dos dados por meio de portas bi-direcionais.

Neste trabalho, foi utilizado o canal 1 (CH1A da figura 6.2) da placa em questão, no modo de operação DH-485, que possibilitou trabalhar com entradas e saídas remotas, ou seja, entradas e saídas comandadas e monitoradas remotamente. O cabo que conecta o computador industrial, através da placa de comunicação, à esteira, entra diretamente pela estação de entradas e saídas número 7, segundo a figura 2.1 e a partir deste ponto todas as estações estão interligadas entre si.

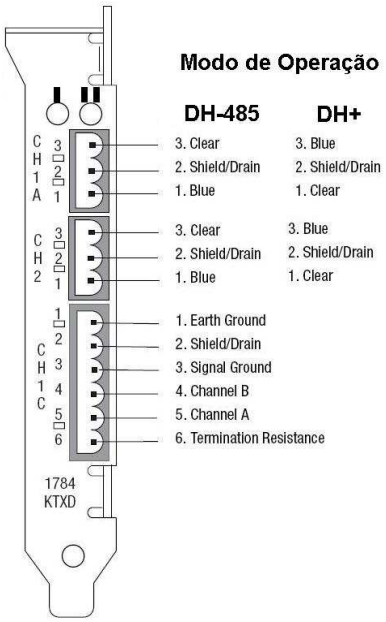


Figura 6.2 – Conexões da placa 1784-KTXD

No capítulo 3, foram descritos os diversos programas instalados no computador industrial que tornam possível a comunicação, e portanto, a automação da esteira transportadora. Já neste capítulo, até o momento, foi descrito o hardware utilizado para que ocorresse a comunicação entre as partes envolvidas.

É interessante evidenciar como estes softwares se comunicam, e isto fica bem claro com a figura 6.3, que nos mostra também as entradas e saídas de dados e conexões externas[15].

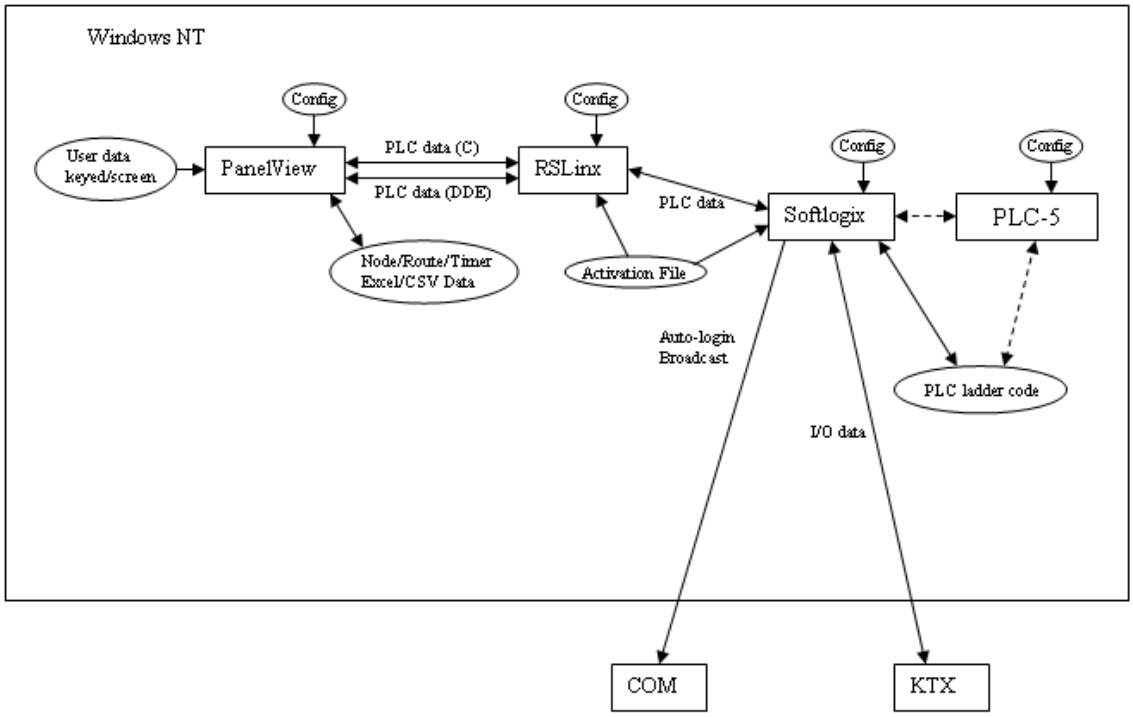


Figura 6.3 – Comunicação entre os diversos softwares e conexões externas

CAPÍTULO 7

RESULTADOS

Atualmente, a automação industrial é muito aplicada visando a melhora da produtividade e qualidade dos processos considerados repetitivos, estando presente no cotidiano das empresas, para apoiar conceitos de produção, tais como os sistemas flexíveis de manufatura e até mesmo o famoso sistema Toyota de produção.

Este trabalho teve o intuito de automatizar uma esteira transportadora, pertencente à uma célula de reciclagem e após os estudos realizados, notou-se que para a programação seria melhor utilizar linguagem Ladder.

Visto que tem-se agora uma noção geral sobre a esteira, o robô e cada um dos seus componentes em particular, cabe a partir deste momento, demonstrar os resultados obtidos com a célula de reciclagem, evidenciando como o sistema reage de acordo situações determinadas com a programação executada.

Baseando-se na figura 2.1, temos o motor principal da esteira (6), acoplado ao redutor (13), o que faz com que as correias da esteira girem continuamente com velocidade constante no sentido de 5 para 1. Isso ocorre pois assim os pallets podem ser levados de uma estação até a próxima e vale notar que é possível elevar todas as estações e manter o pallet sobre ela até que a próxima esteja vaga. Basicamente, existe um intertravamento de cada estação com a próxima (de número inferior), em direção ao robô. Inicialmente, não havendo nenhum pallet no interior do robô, o mesmo permanece na sua posição de descanso e assim sendo, o sensor de descanso(10) do mesmo está ativado.

Um caso que deixa explícito seu funcionamento é aquele em que temos apenas um pallet sobre a esteira, localizado na estação 5 e isso ativa o sensor de presença dessa estação.

Como nenhum sensor de presença das próximas estações está ativado, a estação 5 libera o pallet para ir em direção a estação 4. Como foi dito anteriormente, cada estação está intertravada diretamente com a de número inferior (a número 3 por exemplo, espera um sinal da número 2 dizendo que a mesma está livre, para liberar o pallet).

Visto que nenhuma das próximas estações contêm nenhum pallet, o mesmo segue diretamente até a estação 2. Neste local, temos um intertravamento com o sensor de presença do robô, o sensor de movimentação da estação 1 e o sensor de descanso do robô, ou seja, se o primeiro não estiver ativado, o segundo desativado e

o terceiro ativado, isto indica que a estação 2 pode enviar o pallet para a estação 1 e posteriormente para dentro do robô.

Esta parte é feita através de pistões pneumáticos, que levantam a estação e embaixo da mesma existe um motor(assim como em todas as outras estações), que faz girar uma correia perpendicular a do eixo principal da esteira e com isso o pallet entra no robô.

É importante notar que, enquanto o pallet estiver dentro do robô e alguma atividade esteja sendo executada nele, nenhum outro pode chegar até a estação 1, ficando retidos até o mais próximo dela, que no caso é a estação 2.

Assim que encerra-se tal atividade, o braço do robô retorna para a posição de descanso(10), o pallet volta para a estação 1 e prossegue no eixo principal da esteira, liberando o lugar para que os antecessores(das estações 2, 3, 4 e 5) possam seguir adiante e passar pela desmontagem.

Portanto, notamos que o ciclo de trabalho da célula de reciclagem é simplificadamente demonstrada da maneira como está descrito na figura 7.1:

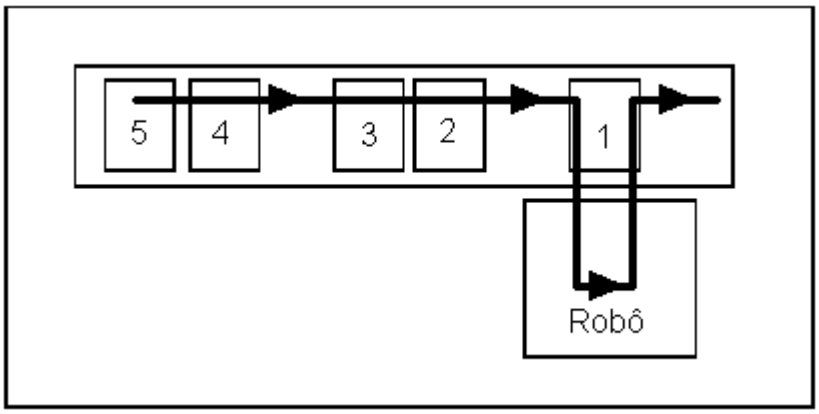


Figura 7.1 – Rota dos pallets na esteira e no robô

Depois que o robô executa a desmontagem do aparelho celular e o pallet sai pela estação 1, o mesmo deveria seguir para algum outro tipo de equipamento em sequência à esteira. No caso desse trabalho, como não existe esse próximo equipamento, o pallet cairia no pavimento e poderia se romper. Para que isso não acontecesse, foi adaptada uma trava mecânica, ou seja, uma chapa metálica colocada depois da estação número 1 que faz com que o pallet fique retido naquele local.

Foi elaborado um algoritmo para facilitar a confecção do programa em linguagem ladder. Este algoritmo fica bem definido com o fluxograma da figura 7.2, que mostra todos os intertravamentos existentes entre as estações, seus motores, sensores, válvulas e o robô.

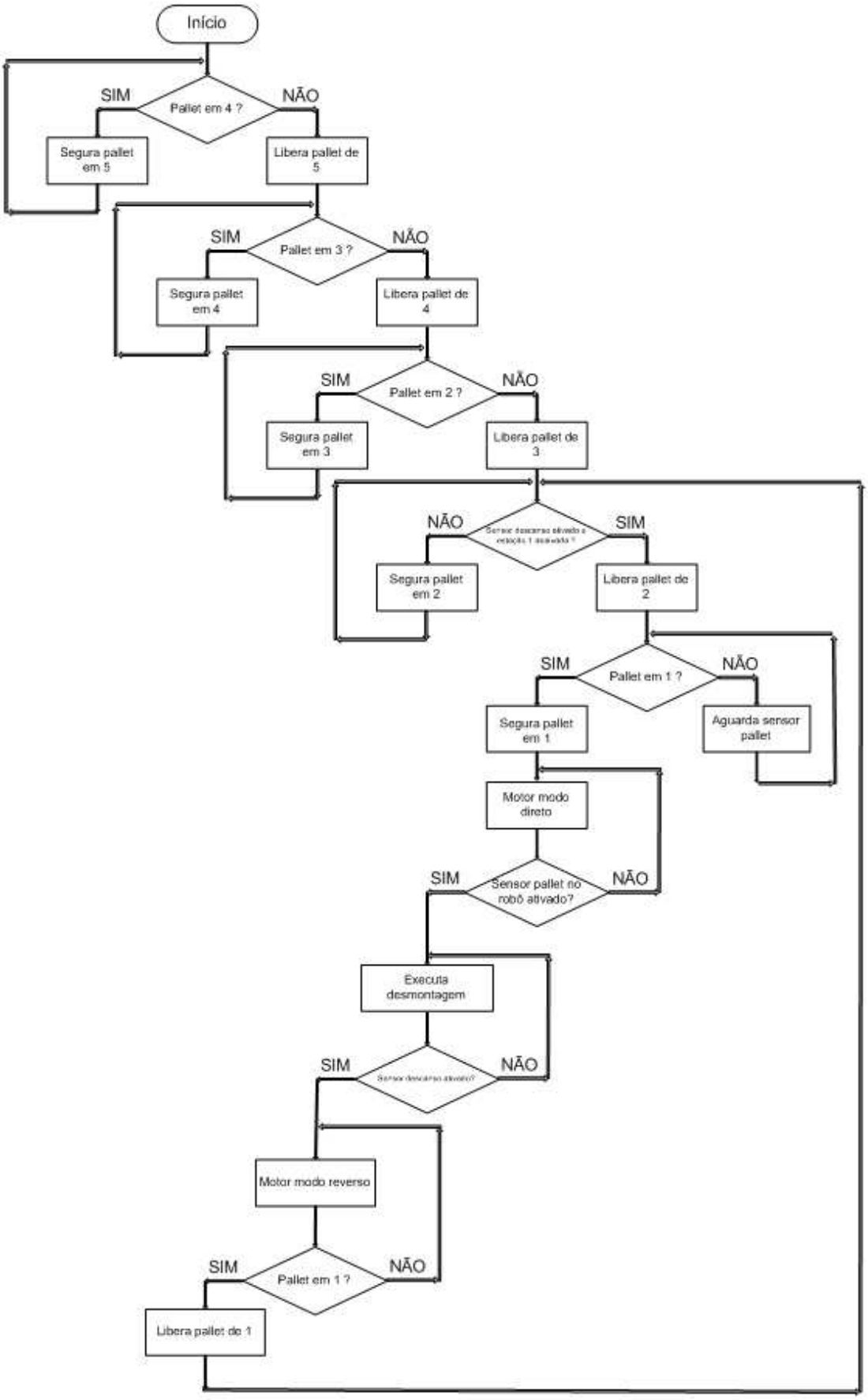


Figura 7.2 – Fluxograma do algoritmo utilizado na programação ladder

Utilizando-se do algoritmo mostrado na figura 7.2, dos conhecimentos prévios de automação industrial e tomando-se como base alguns programas existentes no computador industrial[16], obteve-se como resultado principal a programação em ladder, que automatiza a esteira e realiza o interfaceamento da mesma com o robô. O código ladder será mostrado a seguir, evidenciando cada parte da programação e descrevendo as partes envolvidas nas linhas do referido programa.

Inicialmente, deve-se lembrar do que foi dito no capítulo 4, a respeito do mapeamento das entradas e saídas, pois aqueles endereços foram utilizados para a programação, sendo que os mesmos representam cada equipamento(que atua ou recebe atuação) presente na esteira e, portanto, foram necessários para elaboração da lógica de intertravamento proposta neste trabalho.

Na figura 7.3, nota-se que nas três primeiras linhas do programa, existe um intertravamento entre a estação 5 e a estação 4:

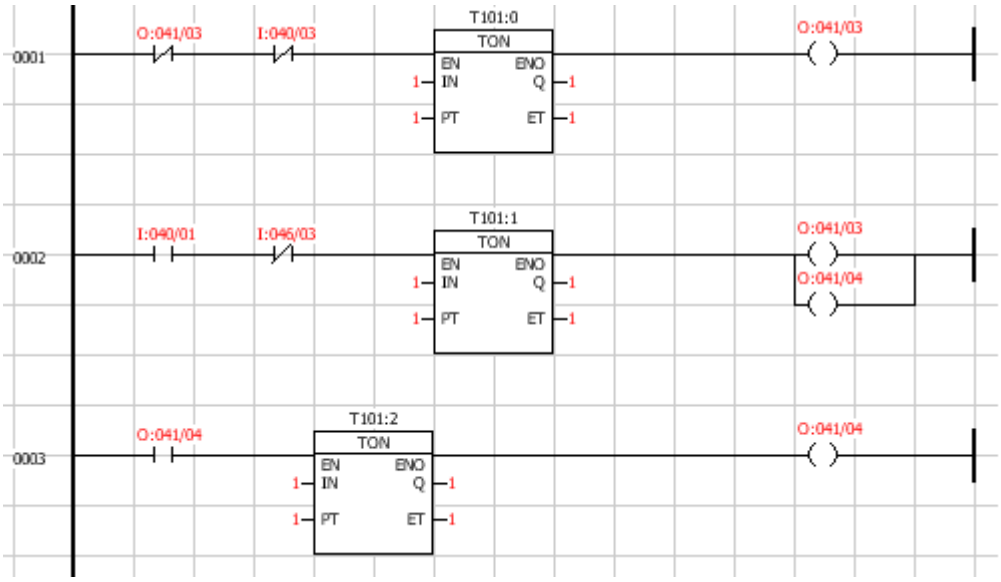


Figura 7.3 – Linhas 1, 2 e 3 do programa ladder do computador industrial

Nesta parte do programa, a estação 5 aguarda sinal de desativação do sensor de movimentação da estação 4, que revela que a estação está livre e que a número 5 pode enviar o pallet para a 4. Uma descrição linha a linha será dada para a figura 7.3 e para figuras 7.4 à 7.7 segue-se a mesma linha de pensamento: na linha 1, quando tem-se um pallet em 5, o pistão eleva a estação e aciona o primeiro contato da segunda linha que só será acionada quando sair o pallet que está em 4, fechando o segundo contato desta linha, fazendo assim com que a estação 5 rebaixe o pistão e o pallet vá para estação 4. Utilizou-se a terceira linha como selo para o recuo do pistão e nota-se que a primeira linha é desativada assim que não existe um pallet em 4.

Na figura 7.4, nota-se que nas linhas 4, 5 e 6 do programa, existe um intertravamento entre a estação 4 e a estação 3. Nesta parte do programa, a estação 4 aguarda sinal de desativação do sensor de movimentação da estação 4, que mostra que o pistão está retraído e que a estação está livre, ou seja, sem nenhum pallet presente. Sendo assim, a estação de número 4 pode enviar o pallet para a de número 3.

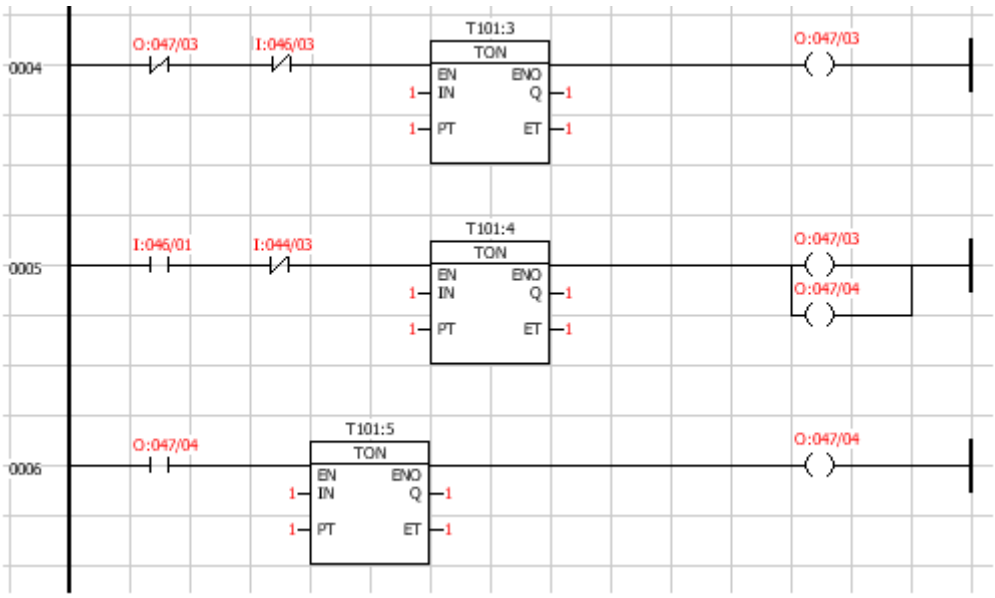


Figura 7.4 – Linhas 4, 5 e 6 do programa ladder do computador industrial

Na figura 7.5, percebe-se que nas linhas 7, 8 e 9 existe um intertravamento entre a estação 3 e 2. Nesta etapa, a estação 3 só libera o pallet quando o sensor de movimentação da estação 2 estiver desativado, mostrando assim que o pistão desta estação está retraído e que a estação está vazia. Portanto, a estação de número 3 pode enviar o pallet para a 2.

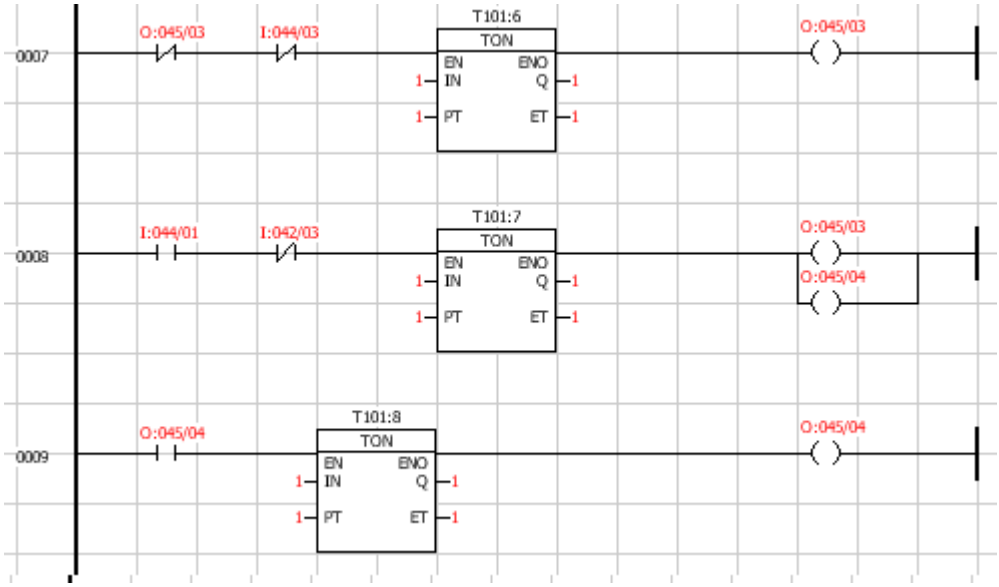


Figura 7.5 – Linhas 7, 8 e 9 do programa ladder do computador industrial

Nota-se, com as figuras 7.6 e 7.7, fica claro que existe um intertravamento entre a estação 1, 2 e o robô, e percebe-se que a programação força a seguinte situação: a estação 2 só libera o pallet que está nela quando o robô tiver terminado de efetuar a desmontagem e o pallet tiver voltado para estação 1 e sido enviado para frente.

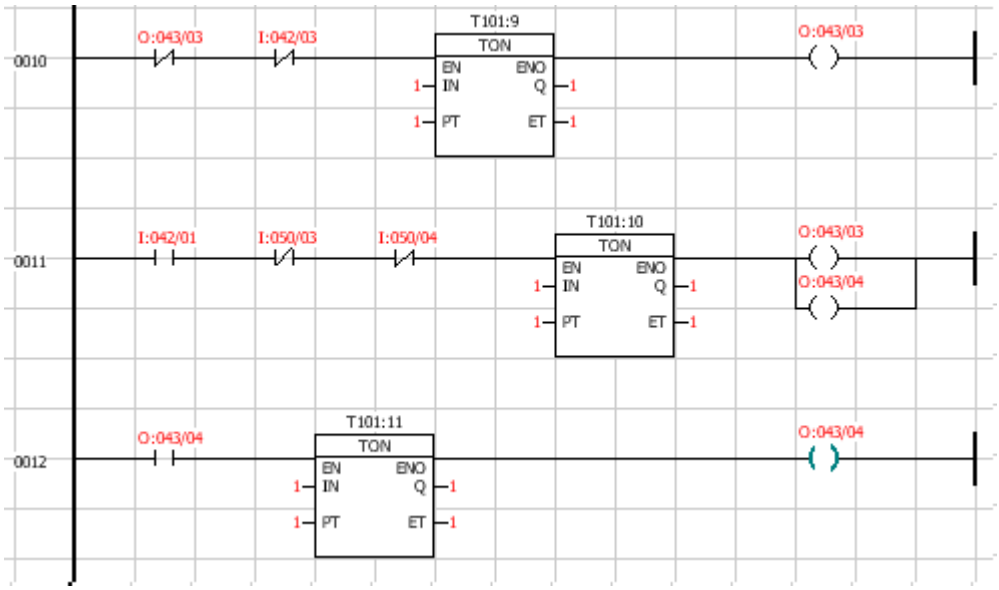


Figura 7.6 – Linhas 10, 11 e 12 do programa ladder do computador industrial

Este intertravamento existe pois, como pode ser visto na figura 7.1, se um pallet estivesse na estação 1 e outro dentro do robô, isto geraria um conflito, visto que um pallet se chocaria com o outro. Logo, este conceito de aguardar o final da

desmontagem, foi de extrema importância para programação ladder da esteira transportadora.

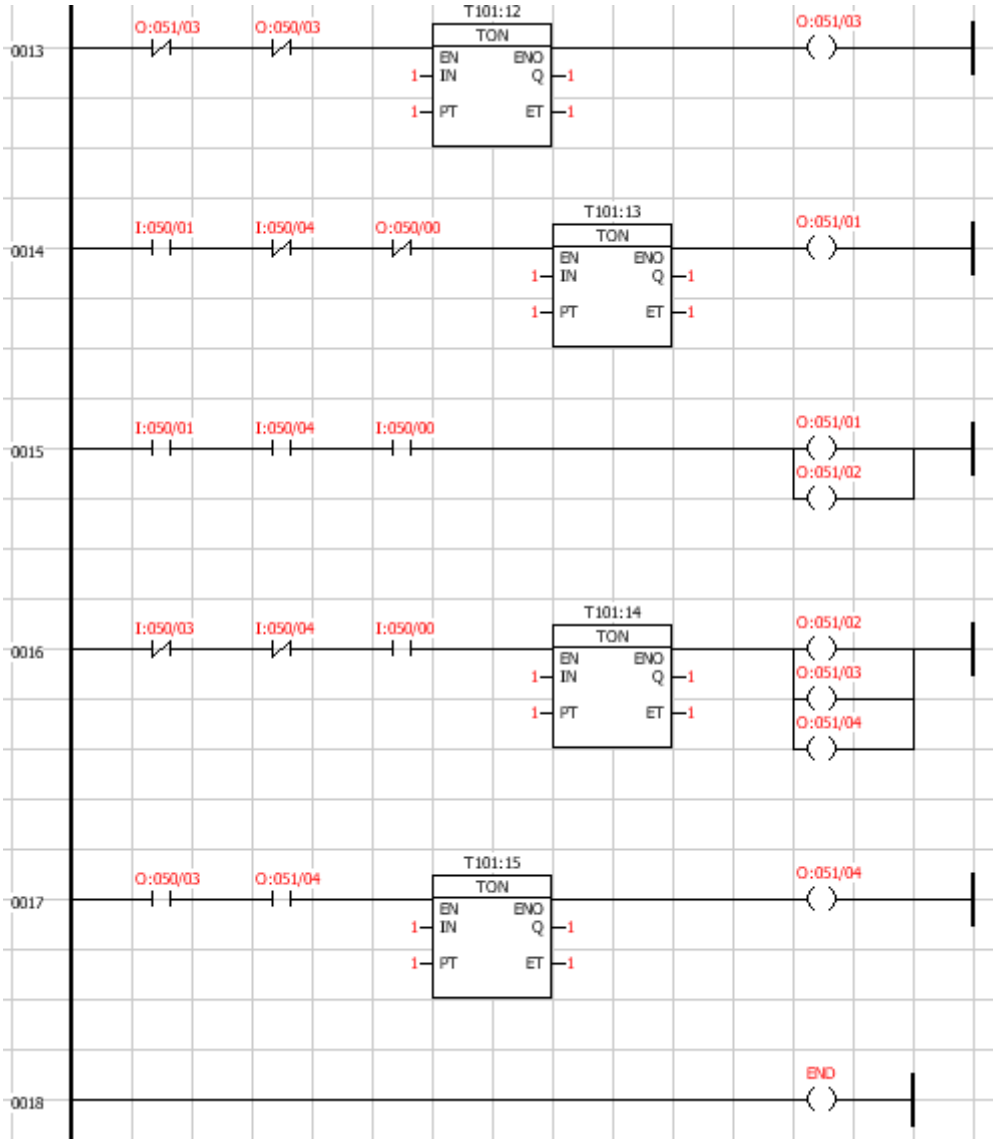


Figura 7.7 – Linhas 13 à 18 do programa ladder do computador industrial

É necessário evidenciar que, tratando-se de lógica ladder, para desempenhar uma mesma função em um equipamento, existem diversas formas de criar um programa e neste trabalho tem-se apenas uma destas maneiras que atendem ao propósito necessário. Porém, para todas essas formas alternativas de programa seria utilizada a mesma lógica de funcionamento para a esteira, que é a de intertravar todas as estações da 5 até a número 2 e aguardar que a desmontagem seja encerrada para liberar o pallet da mesma.

CAPÍTULO 8

CONCLUSÃO

Neste trabalho, foi apresentada uma proposta para implementar, através de lógica ladder, a automação e controle de uma esteira transportadora que integra um sistema de reciclagem de lixo eletrônico.

O projeto consistiu em estudar todos os manuais e diagramas elétricos da esteira e conseguiu-se, a partir destes dados, verificar todos sensores, motores, sistema de alimentação pneumática e elétrica da mesma. Problemas foram encontrados, tanto na parte pneumática quanto na parte elétrica da esteira, devido ao fato da mesma ter ficado um período sem utilização e ter sido movimentada diversas vezes dentro do laboratório e tais problemas foram resolvidos com auxílio dos diagramas presentes no Anexo 1. Porém, com muito critério, todos os problemas foram solucionados e a célula de reciclagem passou a funcionar perfeitamente, finalizando assim grande parte do objetivo final do projeto.

Com seu funcionamento pleno, foi possível dar início aos passos referentes ao computador industrial. Iniciou-se um estudo sobre os softwares e modo de comunicação entre as partes envolvidas, em paralelo com as disciplinas de graduação que tratavam de automação industrial propriamente dita, e desta maneira, tornou-se possível a programação ladder no computador conectado à esteira. Conclui-se que o programa criado se adequou à proposta deste trabalho, ou seja, o sistema de reciclagem está em funcionamento com uma lógica que deixou a entrada e saída de pallets automática.

Uma das metas deste trabalho, que ficou praticamente implícita, é a questão que toca o desenvolvimento sustentável, isto é, a célula de reciclagem permite que seja reduzida drasticamente a eliminação de aparelhos celulares no meio ambiente, permitindo a sua reutilização em outros aparelhos e evitando assim que danos ambientais irreversíveis sejam causados.

Portanto, fica claro que os objetivos previamente dispostos foram atingidos com eficiência, sendo este projeto totalmente funcional, e confirma-se a possibilidade de que este trabalho seja futuramente implementado para diversas aplicações em indústrias, colaborando assim com o desenvolvimento sustentável.

REFERÊNCIAS BIBLIOGRÁFICAS

[1]http://www.fiec.org.br/iel/bolsaderesiduos/Artigos/Artigo_Lixo_tecnologico.pdf, acessado em: 03/10/2008

[2]http://pga.pgr.mpf.gov.br/boletins/arquivos-de-boletins-2008/lixo-tecnologico-o-que-fazer-com-ele, acessado em 03/02/2009

[3] **Technical Data – RSLinx**, 2002, p.1-3 . Disponível em: http://literature.rockwellautomation.com/idc/groups/literature/documents/td/linx-td001_-en-p.pdf, acessado em 07/03/2009.

[4] **RSLinx Classic Getting Results Guide**, Rockwell Software, 2002, p 19-23. Disponível em: http://literature.rockwellautomation.com/idc/groups/literature/documents/gr/linx-gr001_-en-e.pdf, acessado em 20/03/2009.

[5] **SoftLogix - Setup and Configuration Manual,** Rockwell Software, 1999, p. 6-8. Disponível em: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1784-um003_-en-p.pdf, acessado em 20/03/2009.

[6] **A.I. SERIES PLC-5 PROGRAMMING GUIDE**, Rockwell Software, 1997, p. 45-51 . Disponível em: http://literature.rockwellautomation.com/idc/groups/literature/documents/pm/9399-pm001_-en-p.pdf, acessado em 20/03/2009.

[7] **PanelView Manual - Revision 6.04,** Motorola Software, 2000, p. 13-22. Disponível em: http://www.motorola.com/Business/USEN/Business+Solutions/Industry+Solutions/Manufacturing, acessado em 14/05/2009

[8] MIYAGI, P. E. **Controle programável – fundamentos do controle de sistemas e eventos discretos.** 1ª Edição. São Paulo: Edgar Bluncher, 1996. p. 25.

[9] MICHEL, G. **Programmable logic Controllers – Architecture and Applications**. 1ª Edição. Chichester(UK): John Wiley e Sons, 1990, p. 57.

[10] LEWIS, R. W. **Programming industrial control system using IEC 1131-3 – IEC Control**, IEE Control Eng. Series 50, 1997, p. 36-38.

[11] MORAES, C. C. de, CASTRUCCI, P. de L. **Engenharia de automação industrial.** 1ª Edição. Rio de Janeiro: LTC Editora, 2001. p. 29-47.

[12] **Apostila de PLC – Básica.** 6ª Edição. São Paulo: OMRON, 2004, p. 6-8.

[13] Fundamentos da programação LADDER, p. 3-6. Disponível em: http://www.engprod.ufjf.br/epd_automacao/EPD030_Ladder1.pdf, acessado em 17/05/2009.

[14] **User Manual - 1784–KTx Communicaton Interface Card**. **Mayfield Heights:** Allen Bradley, 1999, p.34-38. Disponível em: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1784-um522_-en-p.pdf, acessado em 17/05/2009.

[15] **6180 Industrial Computers - User Manual**, 1998. p. 53-57. Disponível em: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/6180-um001_-en-p.pdf, acessado em 19/03/2009.

[16] FERNANDES A. M. C., SUGITA F. I. **Automação de Esteira Transportadora em Célula de Manufatura.** 2007, p.28-29. Trabalho de Conclusão de Curso. Universidade de São Paulo.

[17] SKERLOS S. J., BASDERE B. **Environmental and Economic View on Cellular Telephone Remanufacturing.** Berlim, 2003, p. 13-14.

ANEXOS

Anexo 1

Foi necessário, para resolver os problemas existentes na alimentação elétrica da esteira, consultar os diagramas elétricos fornecidos pela Motorolla Manufacturing Systems juntamente com a esteira. Tais diagramas estão a seguir.

A) Desenho número SE20074555 – folhas 2 e 3

Nestes dois desenhos, tem-se uma noção geral da alimentação, assim como também do acionamento de cada estação de entradas e saídas.

B) Desenhos AE20096160 – folhas 1 e 2 e AE20103792 – folha 1

Nestes dois desenhos, tem-se a forma como são alimentados, acionados e protegidos os motores das estações na caixa de distribuição.

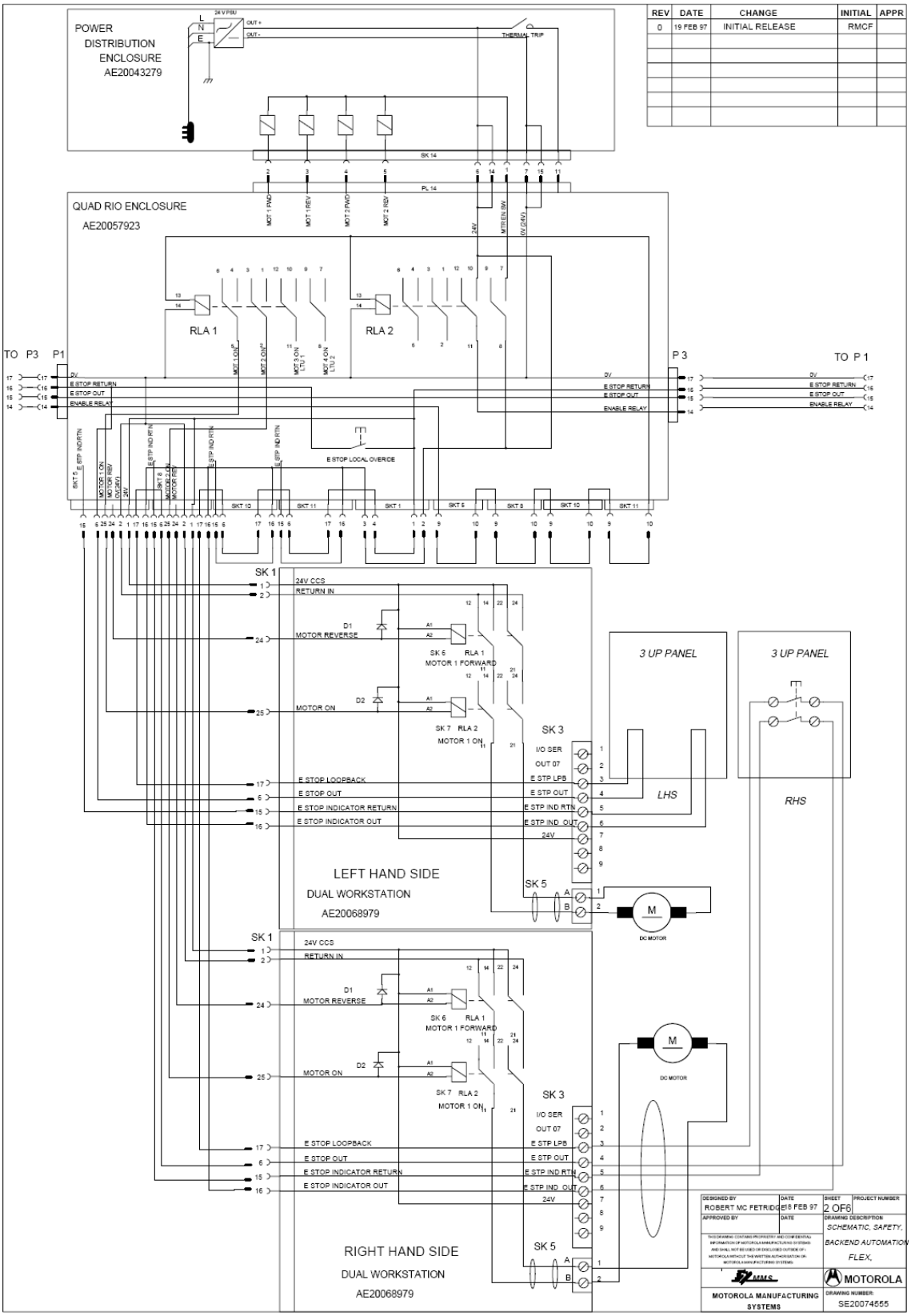
C) Desenho AE20057923 – folha 6

Neste desenhos, tem-se uma vista da frente da estação de entradas e saídas, ficando evidente as Flex I/O, relés e outros diversos componentes de alimentação.

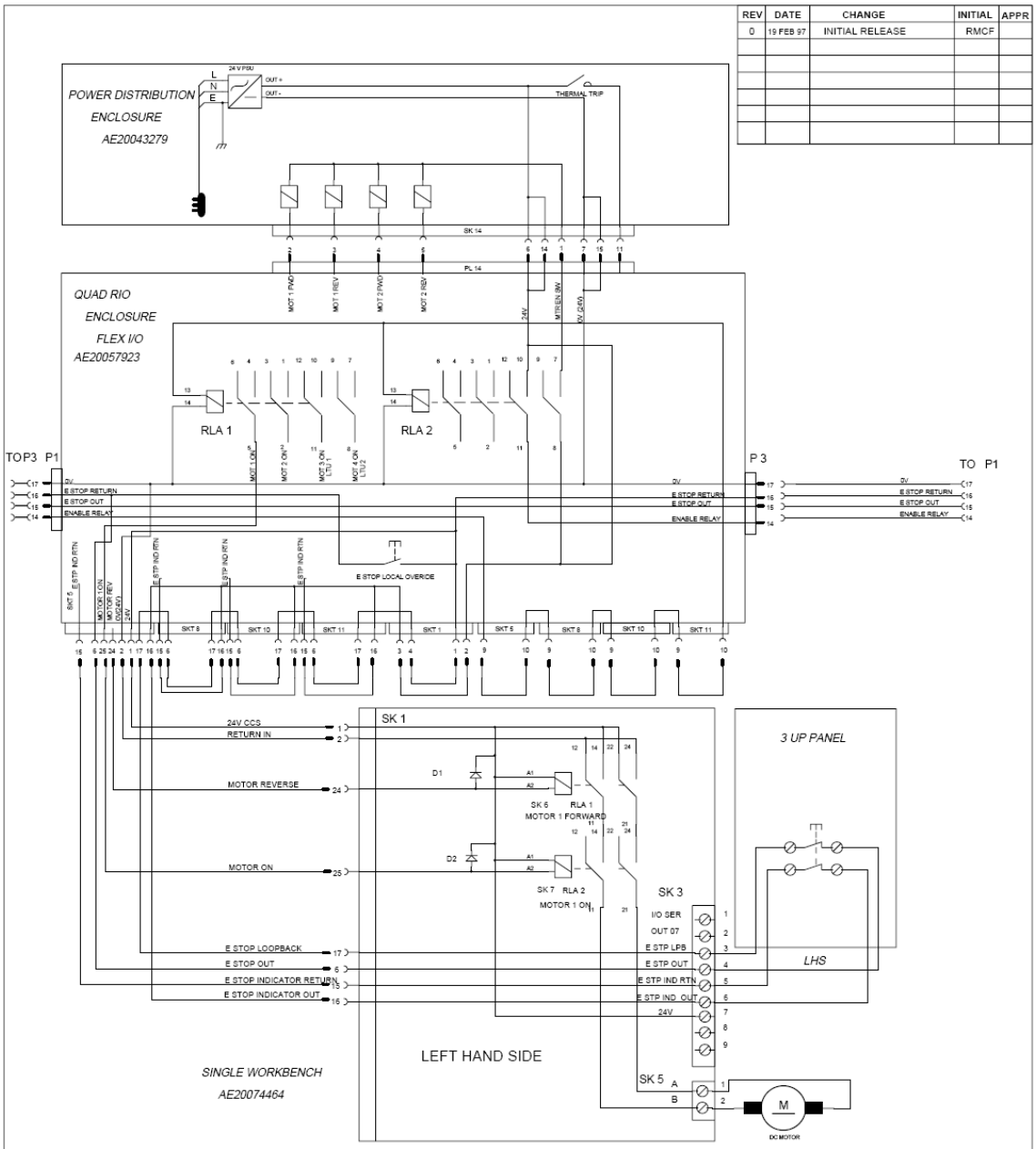
D) Desenho AE20057923 – folha 7

Neste desenhos, tem-se o modo de conexão dos componentes com todas as portas onde pode-se ligar motores, válvulas, chaves e sensores.

A)


POWER DISTRIBUTION ENCLOSURE AE20043279
QUAD RIO ENCLOSURE AE20057923
RLA 1
RLA 2
TO P3 P1
P3
TO P1
E STOP LOCAL OVERIDE
SK 1
SK 3
SK 5
SK 6 RLA 1 MOTOR 1 FORWARD
SK 7 RLA 2 MOTOR 1 ON
MOTOR REVERSE
MOTOR ON
E STOP LOOPBACK
E STOP OUT
E STOP INDICATOR RETURN
E STOP INDICATOR OUT
3 UP PANEL
3 UP PANEL
LHS
RHS
LEFT HAND SIDE
DUAL WORKSTATION
AE20068979
RIGHT HAND SIDE
DUAL WORKSTATION
AE20068979
DC MOTOR
DC MOTOR
REV 0 | DATE 19 FEB 97 | CHANGE INITIAL RELEASE | INITIAL RMCF
DESIGNED BY ROBERT MC FETRIDGE
DATE 18 FEB 97
SHEET 2 OF6
PROJECT NUMBER
APPROVED BY
DATE
DRAWING DESCRIPTION
SCHEMATIC, SAFETY, BACKEND AUTOMATION FLEX,
MOTOROLA MANUFACTURING SYSTEMS
MOTOROLA
DRAWING NUMBER: SE20074555

REV	DATE	CHANGE	INITIAL	APPR
0	19 FEB 97	INITIAL RELEASE	RMCF	



DESIGNED BY	DATE	SHEET	PROJECT NUMBER
R MCFETRIDGE	19 FEB 97	3 OF 6	
APPROVED BY	DATE	DRAWING DESCRIPTION: SCHEMATIC, SAFETY, BACKEND AUTOMATION, FLEX,	

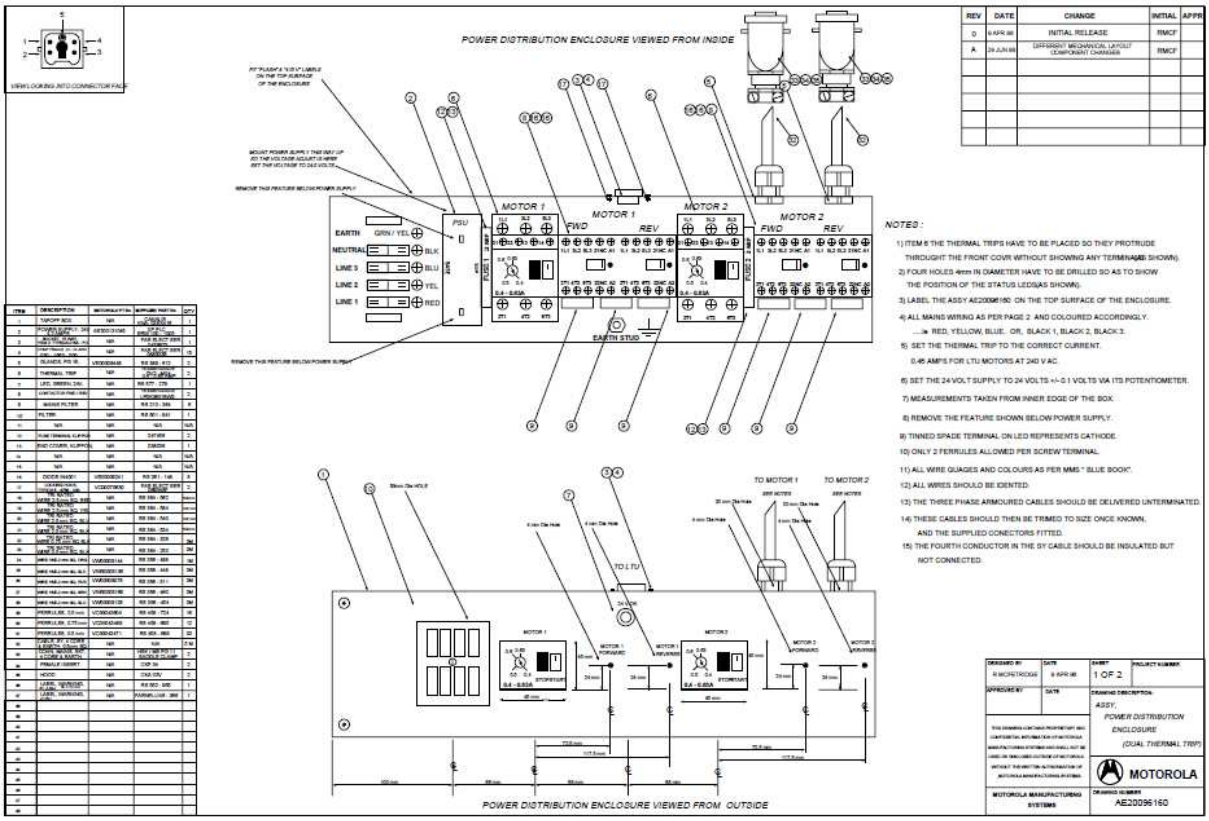
THIS DRAWING CONTAINS PROPRIETARY AND CONFIDENTIAL INFORMATION OF MOTOROLA MANUFACTURING SYSTEMS AND SHALL NOT BE USED OR DISCLOSED OUTSIDE OF MOTOROLA WITHOUT THE WRITTEN AUTHORISATION OF MOTOROLA MANUFACTURING SYSTEMS.

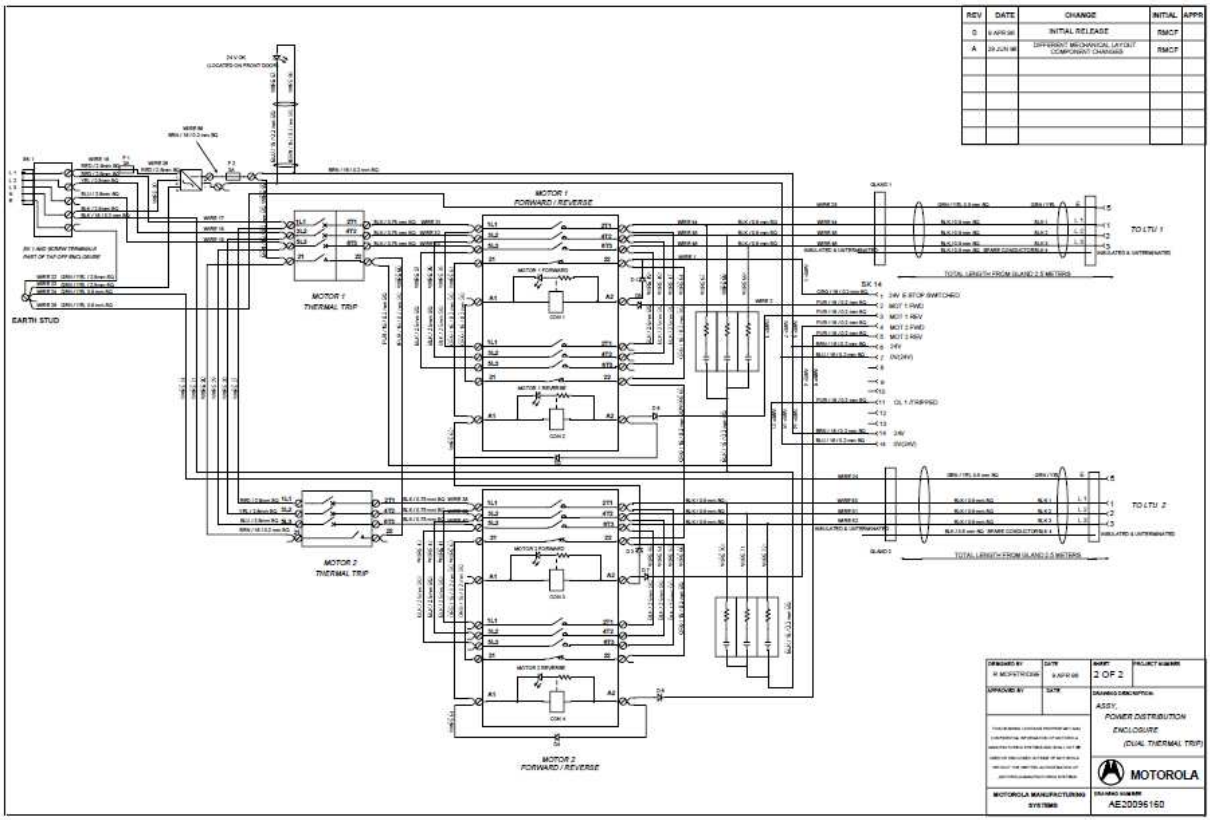
MOTOROLA

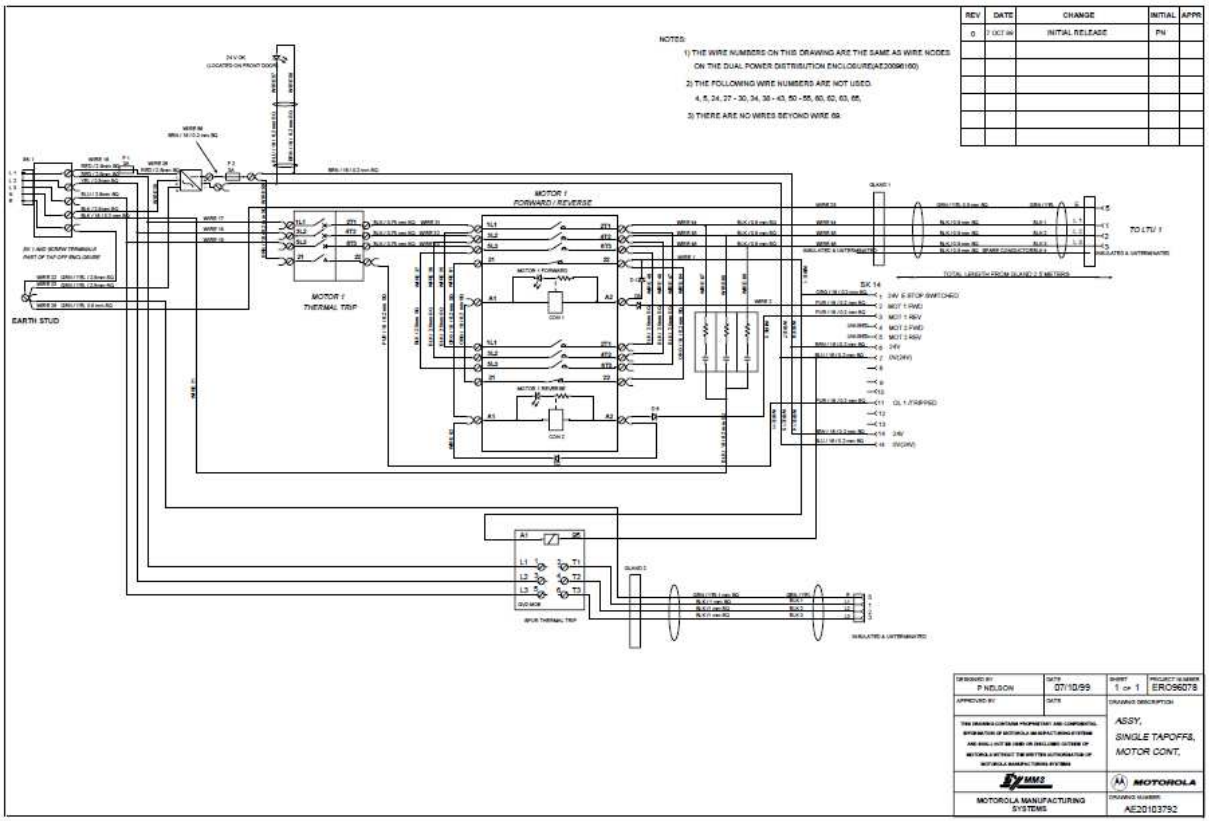
MOTOROLA MANUFACTURING SYSTEMS

DRAWING NUMBER SE20074555

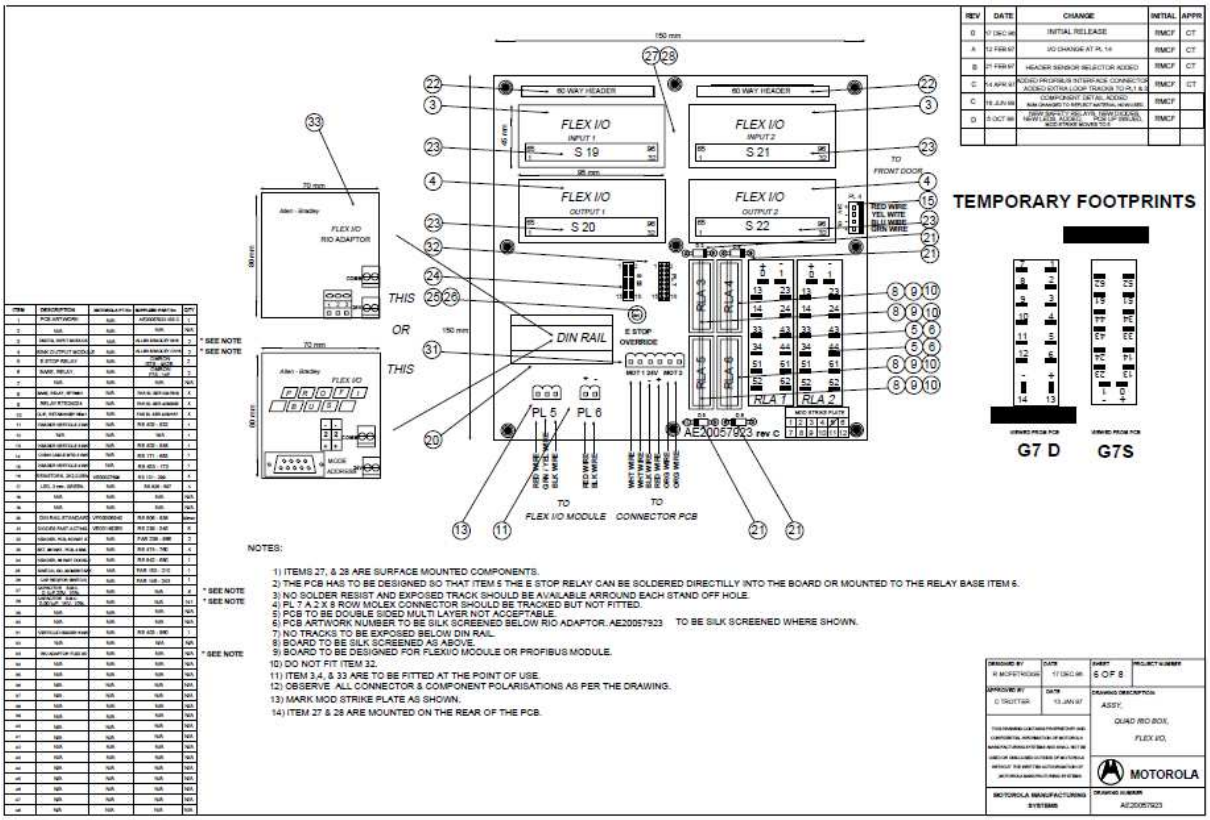
B)


POWER DISTRIBUTION ENCLOSURE VIEWED FROM INSIDE
MOTOR 1
MOTOR 2
FWD
REV
EARTH STUD
NOTES :
TO MOTOR 1
TO MOTOR 2
TO LTU
POWER DISTRIBUTION ENCLOSURE VIEWED FROM OUTSIDE
MOTOROLA
AE20096160

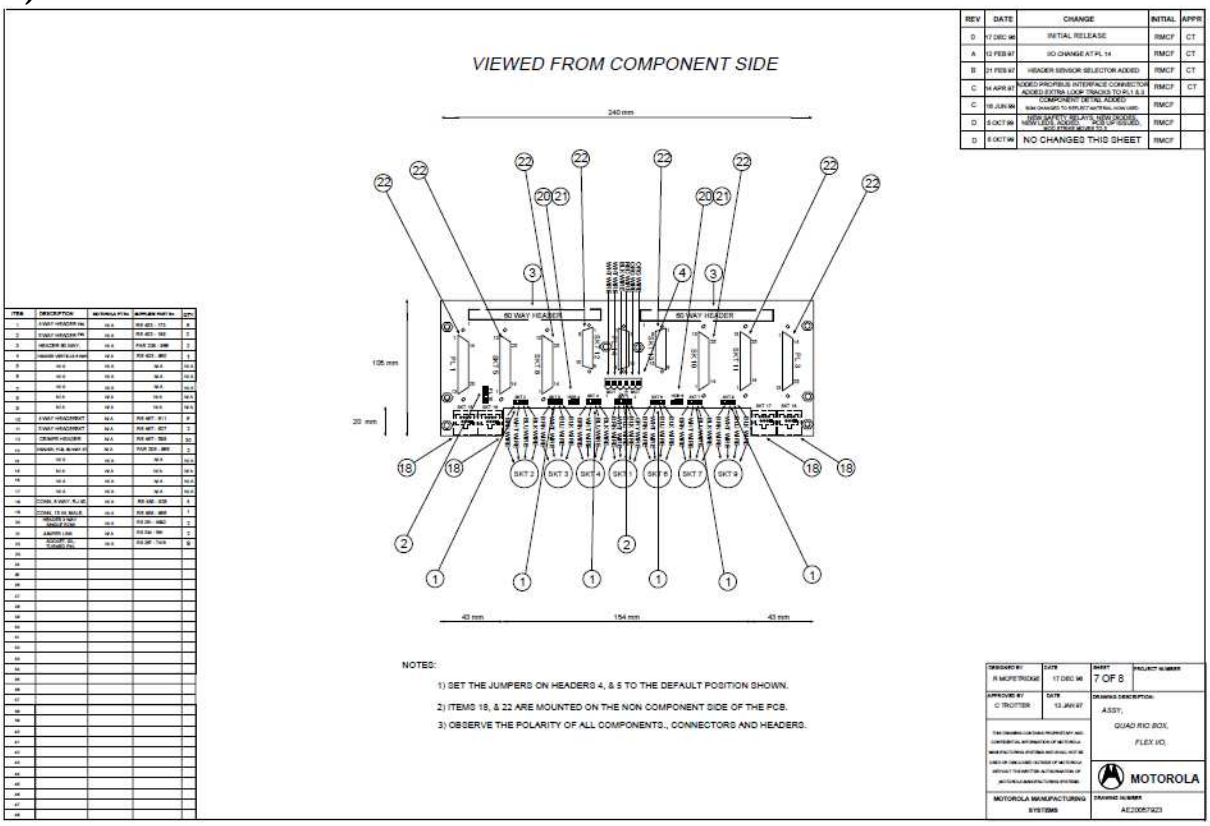

MOTOR 1 FORWARD / REVERSE
MOTOR 1 THERMAL TRIP
MOTOR 2 THERMAL TRIP
MOTOR 2 FORWARD / REVERSE
EARTH STUD
TO LTU 1
TO LTU 2
MOTOROLA
AE20096160


MOTOR 1 FORWARD / REVERSE
MOTOR 1 THERMAL TRIP
EARTH STUD
TO LTU 1
MOTOROLA MANUFACTURING SYSTEMS
AE20103792

C)


60 WAY HEADER
FLEX I/O INPUT 1 S 19
FLEX I/O INPUT 2 S 21
FLEX I/O OUTPUT 1 S 20
FLEX I/O OUTPUT 2 S 22
TO FRONT DOOR
RIO ADAPTOR
THIS OR THIS
DIN RAIL
E STOP OVERRIDE
PL 5
PL 6
RLA 1
RLA 2
RLA 3
RLA 4
RLA 5
RLA 6
AE20057923 rev C
TO FLEX I/O MODULE
TO CONNECTOR PCB
TEMPORARY FOOTPRINTS
G7 D
G7S
NOTES:
1) ITEMS 27, & 28 ARE SURFACE MOUNTED COMPONENTS.
2) THE PCB HAS TO BE DESIGNED SO THAT ITEM 5 THE E STOP RELAY CAN BE SOLDERED DIRECTILLY INTO THE BOARD OR MOUNTED TO THE RELAY BASE ITEM 6.
3) NO SOLDER RESIST AND EXPOSED TRACK SHOULD BE AVAILABLE ARROUND EACH STAND OFF HOLE.
4) PL 7 A 2 X 9 ROW MOLEX CONNECTOR SHOULD BE TRACKED BUT NOT FITTED.
5) PCB TO BE DOUBLE SIDED MULTI LAYER NOT ACCEPTABLE.
6) PCB ARTWORK NUMBER TO BE SILK SCREENED BELOW RIO ADAPTOR. AE20057923 TO BE SILK SCREENED WHERE SHOWN.
7) NO TRACKS TO BE EXPOSED BELOW DIN RAIL.
8) BOARD TO BE SILK SCREENED AS ABOVE.
9) BOARD TO BE DESIGNED FOR FLEXI/O MODULE OR PROFIBUS MODULE.
10) DO NOT FIT ITEM 32.
11) ITEM 3,4, & 33 ARE TO BE FITTED AT THE POINT OF USE.
12) OBSERVE ALL CONNECTOR & COMPONENT POLARISATIONS AS PER THE DRAWING.
13) MARK MOD STRIKE PLATE AS SHOWN.
14) ITEM 27 & 28 ARE MOUNTED ON THE REAR OF THE PCB.
MOTOROLA
MOTOROLA MANUFACTURING SYSTEMS
AE20057923

D)


VIEWED FROM COMPONENT SIDE
240 mm
105 mm
20 mm
60 WAY HEADER
60 WAY HEADER
SKT 2
SKT 3
SKT 4
SKT 5
SKT 6
SKT 7
SKT 8
40 mm
154 mm
40 mm
NOTES:
1) SET THE JUMPERS ON HEADERS 4, & 5 TO THE DEFAULT POSITION SHOWN.
2) ITEMS 18, & 22 ARE MOUNTED ON THE NON COMPONENT SIDE OF THE PCB.
3) OBSERVE THE POLARITY OF ALL COMPONENTS., CONNECTORS AND HEADERS.
REV
DATE
CHANGE
INITIAL
APPR
D 17 DEC 96 INITIAL RELEASE RMCF CT
A 12 FEB 97 I/O CHANGE AT PL 14 RMCF CT
B 21 FEB 97 HEADER SENSOR SELECTOR ADDED RMCF CT
C 14 APR 97 ADDED PROFIBUS INTERFACE CONNECTOR ADDED EXTRA LOOP TRACKS TO PL1 & 2 RMCF CT
C 18 JUN 96 COMPONENT DETAIL ADDED RMCF
D 5 OCT 96 NEW SAFETY RELAYS, NEW DIODES NEW LED, NEW CAPS, NEW RESISTORS, BOS STAND ALONE OLA RMCF
D 5 OCT 96 NO CHANGES THIS SHEET RMCF
DESIGNED BY R MCPETRIDGE
DATE 17 DEC 96
SHEET 7 OF 8
PROJECT NUMBER
APPROVED BY C TROTTER
DATE 13 JAN 97
DRAWING DESCRIPTION: ASSY, QUAD RIO BOX, FLEX I/O.
MOTOROLA
MOTOROLA MANUFACTURING SYSTEMS
DRAWING NUMBER AE20057923