

9.3 (nove e três)



**Universidade de São Paulo**

**Escola Politécnica**

**Departamento de Engenharia Mecatrônica**

**Relatório de Projeto de Formatura**

# ***Sistema de Posicionamento por Cinemática Paralela***

Alunos: Daniel T. B. Moraes    nº USP: 2234186

Marcelo Boczko    nº USP: 2233272

Orientador:    Prof. Tarcísio A. Hess Coelho

São Paulo, 13 de dezembro de 2000.

Daniel Teixeira de Barros Moraes

Marcelo Boczko

## SISTEMA DE POSICIONAMENTO POR CINEMÁTICA PARALELA

Trabalho de formatura apresentado à  
Escola Politécnica da Universidade de São Paulo

Área de Concentração:

Engenharia Mecatrônica

Orientador:

Tarcísio A. H. Coelho

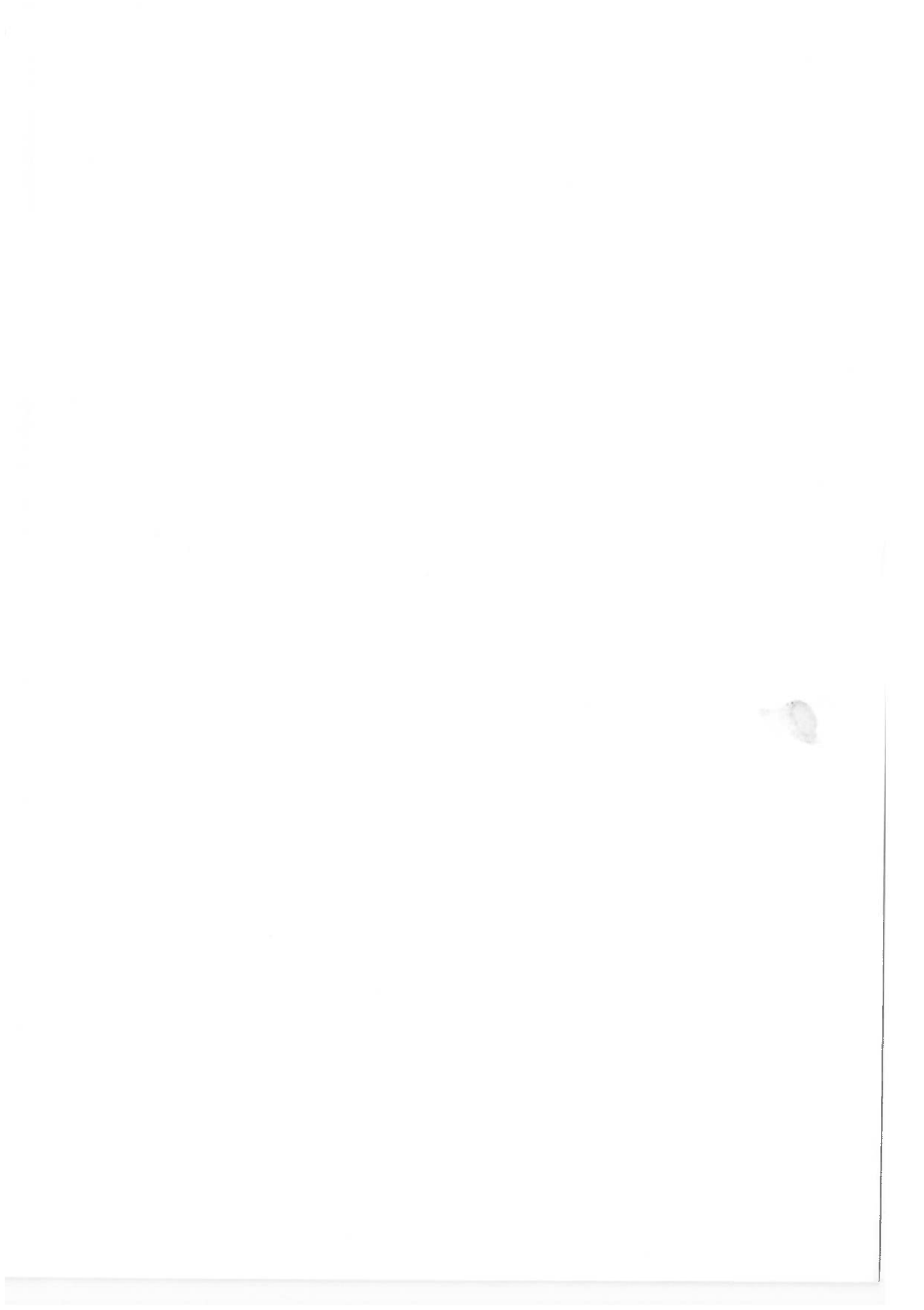
São Paulo

11 - 12 - 2000

**DEDALUS - Acervo - EPMN**



31600005967



## **Dedicatórias**

Marcelo:

Dedico este trabalho ao meu amigo Daniel.

Daniel:

Dedico este trabalho ao Marcelo, aos alunos e colegas que iniciaram este projeto, à meus pais e parentes, aos meus amigos e ao Prof. Tarcísio (apesar de ele se enquadrar no último item) e a todos os que tiverem a paciência de lerem este trabalho.

## Agradecimentos . . .

### **Marcelo:**

Agradeço às pessoas que me ajudaram durante os momentos difíceis no desenvolvimento deste projeto, em especial ao Daniel, amigo de muitas horas e ao Prof. Tarcísio, que sempre compreendeu as dificuldades por que passávamos.

Dedico também à Taís, ao Mateus, ao meu pai e ao Michel, que deram apoio simplesmente por oferecer o ombro, os ouvidos ou por agüentarem o mau humor que cansaço causava.

### **Daniel:**

Agradeço a todos que de uma forma ou de outra me ajudar a sobrepor as dificuldades que me foram apresentadas durante a faculdade, especialmente às dificuldades deste trabalho que não foram poucas. Agradeço ao Prof. Tarcísio pela oportunidade e por todo o apoio e dedicação. Agradeço principalmente aos meus pais por todo o apoio e dedicação e principalmente ao Marcelo, muito mais que um simples companheiro de sala ou de trabalhos infundáveis (como este), um amigo, à Carô, minha irmã e amiga, à Taís e à Claudia, minhas estrelas-guia por **TUDO**.

Em nome de ambos, digo que as palavras são pequenas para expressar nossa gratidão ao corpo técnico que viabilizou este trabalho, ao querido Prof. Tarcísio pelo apoio, pelo auxílio, pelo aprendizado, pela paciência, ao Prof. Marcílio e ao Alex; a todos os funcionários da oficina, da biblioteca e das secretarias. Também aos representantes das empresas parceiras, SEW, Phoenix Contact, Play Corp e Droid pela viabilização deste projeto, em especial ao Anderson, ao Juliano, ao Reinaldo e ao Dimas.

# Índice

|                                                                                     |           |
|-------------------------------------------------------------------------------------|-----------|
| <b>1 - INTRODUÇÃO .....</b>                                                         | <b>13</b> |
| 1.1 - OBJETIVOS .....                                                               | 14        |
| <b>2 - FUNDAMENTAÇÃO TEÓRICA .....</b>                                              | <b>15</b> |
| 2.1 - PLATAFORMA DE STEWART .....                                                   | 15        |
| 2.2 - PLATAFORMA DE STEWART SIMPLIFICADA.....                                       | 18        |
| 2.2.1 - Histórico.....                                                              | 18        |
| 2.2.2 - Aplicações possíveis e orientações para o desenvolvimento em paralelo ..... | 20        |
| <b>3 - PROTÓTIPO .....</b>                                                          | <b>22</b> |
| 3.1 - DIMENSIONAMENTO DOS MOTORES .....                                             | 24        |
| 3.2 - EQUIPAMENTOS DOADOS .....                                                     | 27        |
| 3.2.1 - Motores elétricos trifásicos .....                                          | 27        |
| 3.2.2 - Conversores de frequência .....                                             | 28        |
| 3.2.3 - Resistores de Frenagem .....                                                | 31        |
| 3.2.4 - Placa de aquisição e comunicação de dados.....                              | 31        |
| 3.2.5 - Software de implementação.....                                              | 32        |
| 3.3 - SUBSISTEMA MECÂNICO .....                                                     | 34        |
| 3.4 - SUBSISTEMA DE COMUNICAÇÃO E AQUISIÇÃO DE DADOS (INTERBUS).....                | 36        |
| 3.4.1 - Dados de Processo (Process Data – PD's) .....                               | 38        |
| 3.4.2 - Palavras de Controle (Control Words) .....                                  | 39        |
| 3.4.3 - Palavras de Status.....                                                     | 44        |
| 3.4.4 - Variáveis Lógicas de Programação.....                                       | 48        |
| 3.5 - SUBSISTEMA DE ACIONAMENTO .....                                               | 50        |
| 3.5.1 - Painel Frontal do MD.....                                                   | 52        |
| 3.5.2 - Placas de comunicação para protocolo Interbus (DF111A).....                 | 56        |
| 3.6 - SUBSISTEMA DE CONTROLE .....                                                  | 58        |

|                                                           |           |
|-----------------------------------------------------------|-----------|
| 3.7 - SUBSISTEMA ELÉTRICO .....                           | 59        |
| <b>4 - MÁQUINA DE FEIRA.....</b>                          | <b>63</b> |
| 4.1 - MODO DEMO (DEMONSTRATIVO) .....                     | 64        |
| 4.2 - MODO JOGO.....                                      | 66        |
| <b>5 - SISTEMA DE POSICIONAMENTO.....</b>                 | <b>70</b> |
| <b>6 - RESULTADOS OBTIDOS.....</b>                        | <b>71</b> |
| <b>7 - CONCLUSÕES E PERSPECTIVAS FUTURAS.....</b>         | <b>72</b> |
| <b>8 - REFERÊNCIAS BIBLIOGRÁFICAS .....</b>               | <b>73</b> |
| <b>LISTA DOS APÊNDICES.....</b>                           | <b>74</b> |
| A - Tabela de códigos de falha dos MD's                   |           |
| B – Código-fonte do programa do PC em Visual Basic        |           |
| C – Códigos-fonte dos programas dos MD's em IPos Plus®    |           |
| D – Tabelas de Descrição do Projeto de Fiação             |           |
| E – Indicações de Montagem para o <i>Remote Bus Cable</i> |           |
| F – Desenhos de Fabricação                                |           |
| G – FieldBus                                              |           |

# Lista de Figuras

|                                                                                                                                              |    |
|----------------------------------------------------------------------------------------------------------------------------------------------|----|
| FIGURA 1 - EXEMPLO DE PLATAFORMA DE STEWART. ....                                                                                            | 16 |
| FIGURA 2 - PLATAFORMA DE STEWART SIMPLIFICADA EM JANEIRO DE 2000. ....                                                                       | 19 |
| FIGURA 3 - SUBSISTEMAS COMPONENTES .....                                                                                                     | 23 |
| FIGURA 4 - INTER-RELACIONAMENTOS ENTRE SUBSISTEMAS .....                                                                                     | 24 |
| FIGURA 5 - MOVIMENTO ADOTADO COMO REFERÊNCIA PARA DETERMINAÇÃO DE ESPECIFICAÇÕES DOS MOTORES                                                 | 25 |
| FIGURA 6 - VISTA EXPLODIDA DO MOTOR COM DETALHE DO ENCODER. ....                                                                             | 28 |
| FIGURA 7 - FOTO DE UM MOVIDRIVE DO MODELO INDICADO. ....                                                                                     | 29 |
| FIGURA 8 - PLACA DFII1A COM SEUS CONECTORES E LED's. ....                                                                                    | 30 |
| FIGURA 9 - EXEMPLO DE DADOS QUE TRANSITAM A VIA INTERBUS. ....                                                                               | 32 |
| FIGURA 10 - ESTRUTURA DOS DRIVERS DE INTERBUS E INTERFACES. ....                                                                             | 33 |
| FIGURA 11 - DETALHE DO ALOJAMENTO. OS PARAFUSOS NA SOMBRA PERMITEM<br>RETIRAR O MOTOR SEM DESMONTAR A PARTE SUPERIOR QUE CONTEM O FUSO. .... | 34 |
| FIGURA 12 - DETALHE DO FUSO COM AMORTECEDOR DE FIM DE CURSO. ....                                                                            | 35 |
| FIGURA 13 - FIGURA ESQUEMÁTICA DOS ELEMENTOS QUE COMPÕE O SUB-SISTEMA MECÂNICO. ....                                                         | 36 |
| FIGURA 14 - EXEMPLO DE VARIÁVEIS DE PROCESSO (PD's) .....                                                                                    | 38 |
| FIGURA 15 - TIPOS DE <i>CONTROL WORD</i> . ....                                                                                              | 40 |
| FIGURA 16 - PRIORIDADE DE BITS NA <i>CONTROL WORD</i> . ....                                                                                 | 42 |
| FIGURA 17 - GRÁFICO DO MD1 EM SITUAÇÃO DE <i>HOLD CONTROL</i> SOB CARGA VARIÁVEL. ....                                                       | 43 |
| FIGURA 18 - TIPOS DE <i>STATUS WORDS</i> . ....                                                                                              | 45 |
| FIGURA 19 - UNIDADE DE ACIONAMENTO .....                                                                                                     | 51 |
| FIGURA 20 - GRÁFICO DE COMPARAÇÃO DO TEMPO DE RESPOSTA DOS MODOS VFC E CFC. ....                                                             | 52 |
| FIGURA 21 - PAINEL FRONTAL DO MD .....                                                                                                       | 53 |
| FIGURA 22 - PLACA DE INTERFACE MD – INTERBUS .....                                                                                           | 57 |
| FIGURA 23 - ESQUEMA ELÉTRICO DE INTEGRAÇÃO DOS DIVERSOS SUB-SISTEMAS. ....                                                                   | 59 |
| FIGURA 24 - CONECTOR DE POTÊNCIA UTILIZADO PARA LIGAR O GABINETE À MESA. ....                                                                | 60 |
| FIGURA 25 - VISTAS DO CONVERSOR DE PROTOCOLO RS485 PARA RS232. ....                                                                          | 61 |

|                                                    |    |
|----------------------------------------------------|----|
| FIGURA 26 - FOTO DA MÁQUINA DE FEIRA. ....         | 63 |
| FIGURA 27 - GRÁFICO DO MD4 DURANTE MODO DEMO. .... | 66 |
| FIGURA 28 - MESA DE JOGO .....                     | 67 |

## Lista de Tabelas

|                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------|----|
| TABELA 1 - MODOS DE CONVERSÃO DE VELOCIDADE E POSIÇÃO.....                                            | 39 |
| TABELA 2 - FUNÇÕES BIT A BIT DAS PALAVRAS DE CONTROLE.....                                            | 41 |
| TABELA 3 - CONFIGURAÇÃO DOS BORNES VIRTUAIS (BITS MAIS SIGNIFICATIVOS) DOS <i>CONTROL WORDS</i> ..... | 44 |
| TABELA 4 - ESTADOS BIT A BIT DAS PALAVRAS DE <i>STATUS</i> .....                                      | 46 |
| TABELA 5 - CONFIGURAÇÃO DOS BORNES VIRTUAIS (BITS MAIS SIGNIFICATIVOS) DOS <i>STATUS WORDS</i> .....  | 48 |
| TABELA 6 - VARIÁVEIS LÓGICAS DE PROGRAMAÇÃO.....                                                      | 48 |
| TABELA 7 - COMPARAÇÃO ENTRE MODOS VFC E CFC.....                                                      | 51 |
| TABELA 8 - OPÇÕES DE OPERAÇÃO CONFORME <i>DISPLAY</i> DE 7 SEGMENTOS.....                             | 54 |
| TABELA 9 - CONECTOR X10.....                                                                          | 55 |
| TABELA 10 - CONECTOR X13.....                                                                         | 55 |

## Resumo

Mecanismos baseados em Plataformas de Stewart têm sido alvo de pesquisas em universidades do mundo todo e, recentemente, têm viabilidade técnica e vêm sendo utilizados em projetos de empresas multinacionais, como Siemens e Guiddings & Lewis (Variac).

Este trabalho baseia-se em uma plataforma de Stewart simplificada, cujo projeto foi inicialmente elaborado por alunos desta Escola, sob orientação do professor Tarcísio A. H. Coelho, em anos anteriores.

O projeto e implementação atuais incorporam um sistema mecânico bastante semelhante ao projeto inicial e que funciona conjuntamente com um sistema de controle e um de acionamento, constituído por 4 conversores de frequência programáveis e quatro motores trifásicos.

De forma a viabilizar este projeto de pesquisa e desenvolvimento, devido aos altos custos dos conversores de frequência e do sistema de aquisição e comunicação de dados (Interbus), foi formalmente acordado entre os alunos, o orientador, a empresa fornecedora dos conversores (SEW) e a empresa fornecedora do sistema Interbus (Phoenix Contact), a realização de um plano conjunto de desenvolvimento para o sistema, o que serviu inclusive para a demonstração em feiras e exposições relacionadas com mecânica, automação e elétrica / eletrônica.

## **Abstract**

Stewart Platform based mechanisms have been researched in Universities around the world. Recently they have gotten developed technical background and have been used in projects at large corporations such Siemens and Guiddings & Lewis (Variac).

This work is based on a simplified Stewart Platform, being a continuation of an earlier paper done by students of the Politechnical School of São Paulo, also oriented by Ph. D. Tarcísio A. H. Coelho in previous years.

Nowadays, it is composed of a mechanical system (very much alike the previous one) that works with a control and actuators (4 programmable frequency inverters and 4 trifasic electrical motors).

To make the project possible, considering the high costs of parts such as the frequency inverter and the control and data acquisition system (Interbus), an agreement was made among the authors, the oriented students and the donating companies (SEW donating the frequency inverters and the motors and Phoenix Contact donating the Interbus system facilities) in order to make a development plan to design a system to be shown at technical fairs and congresses.



## 1 - Introdução

O trabalho descrito no presente relatório é referente ao projeto derivado de um mecanismo elaborado por um aluno desta Escola, André Mitsuyoshi Oura, como trabalho de formatura do ano de 1998, sob orientação do professor Tarcísio A. H. Coelho. O mecanismo foi continuado como trabalho de formatura de alunos de 1999 (Danielle Morita, Fábio Fukunaga, Maurício Dias Menezes Mazza e Rafael Salla Paulino), também sob orientação do professor Tarcísio.

No ano de 2000, este mecanismo foi aprimorado, tendo suas características mecânicas melhor formuladas e alteradas, de maneira a suportar maiores esforços. Foram ainda acrescentados ao mecanismo um sistema de controle e um conjunto de acionadores, capacitando o mecanismo de movimento autônomo e controlado.

O mecanismo projetado é capaz de gerar o posicionamento de um sistema de referência com quatro graus de mobilidade, utilizando-se para isto do conceito simplificado de Plataforma de Stewart.

Plataforma de Stewart é um mecanismo de posicionamento capaz de dar seis graus de mobilidade a um sistema de referência. Mecanismos baseados em plataformas de Stewart vem sendo largamente estudados e, de maneira simplificada, utilizados.

Com a inclusão dos sistemas de controle e de acionamento, acrescentados ao aprimoramento das características mecânicas de implementação do mecanismo, as possibilidades de utilização deste sistema de posicionamento foram largamente ampliadas, sendo previstos usos em sistemas de posicionamento de precisão, sistemas de posicionamento de peças para usinagem e simuladores.

O mecanismo, da maneira como é implementado, possui cadeia cinemática fechada, o que atribui a ele características que serão posteriormente discutidas.

A viabilidade de implementação dos sistemas de controle e de acionamento foram consolidadas através de um acordo de cooperação entre a USP, a SEW e a Phoenix Contact. Neste contrato, a empresa SEW doou quatro motores, quatro conversores de frequência e periféricos, além de extenso apoio técnico; a empresa Phoenix Contact doou módulos de comunicação Interbus, prestou apoio técnico e forneceu um curso sobre o protocolo Interbus e sobre a implementação do sistema de comunicação no referido projeto. Em contrapartida, a USP, através do Professor Tarcísio e dos alunos por ele orientados, comprometia-se a desenvolver o sistema de modo a torná-lo utilizável como objeto de exposição em feiras especializadas.

## 1.1 - Objetivos

Este trabalho foi desenvolvido para atender a dois principais objetivos:

- Criação de um sistema de posicionamento utilizável em feiras e exposições, com intuito de divulgar e promover os módulos de controle, comunicação e de acionamento do mesmo;
- Elaboração de um sistema de posicionamento, com acionamento automático, visando o uso didático e atendendo perspectivas de utilização em sistemas de posicionamento e de movimentação de aplicações mais amplas, desenvolvendo os conceitos para uma possível futura utilização comercial.

## 2 - Fundamentação Teórica

### 2.1 - Plataforma de Stewart

A plataforma de Stewart consiste de um mecanismo de cinemática paralela, em contraste com a cinemática serial, comum em robôs. A cinemática serial caracteriza-se por ter os movimentos de seus módulos cinemáticos dependentes da posição, velocidade e aceleração (lineares e angulares) dos módulos anteriores. A cinemática paralela dispõe seus módulos cinemáticos de maneira independente, ou seja, a movimentação ou posicionamento de um módulo cinemático independe dos demais.

A disposição paralela dos módulos cinemáticos, constituindo uma cadeia cinemática fechada, confere uma maior estabilidade e rigidez mecânica, além de maior precisão e acurácia de posicionamento. Tem a desvantagem, no entanto, de possuir pontos de singularidade, nos quais o controle é impossível ou instável.

A Plataforma de Stewart é composta por seis atuadores que ligam uma base estacionária a uma plataforma móvel. Sendo os movimentos dos atuadores completamente desvinculados, a plataforma móvel pode ser posicionada com seis graus de liberdade.

Uma aplicação possível para a Plataforma de Stewart é apresentada na Figura 1, onde seis atuadores lineares, vinculados à base fixa e à plataforma móvel, são responsáveis pelo posicionamento desta.

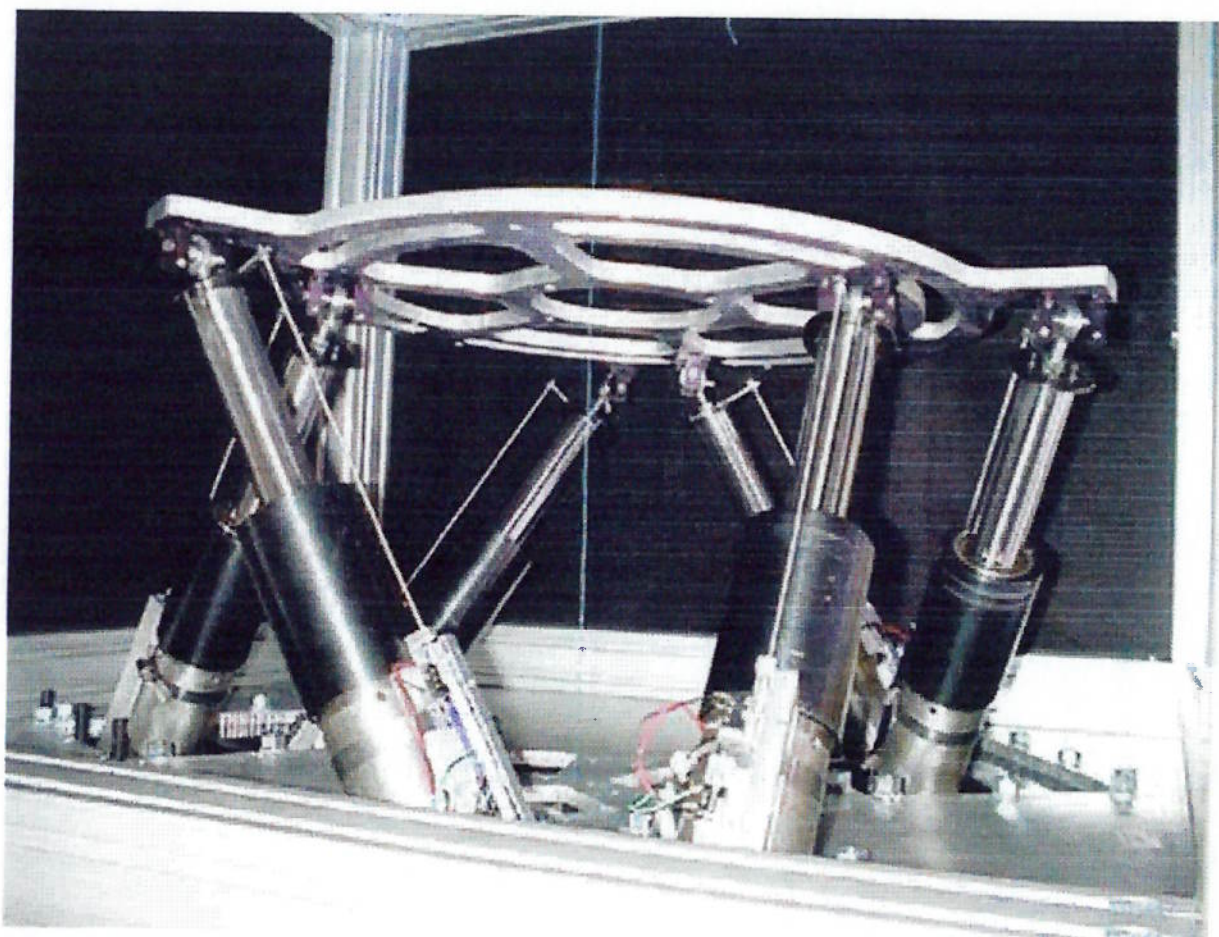


Figura 1 - Exemplo de plataforma de Stewart.

O uso de uma Plataforma de Stewart, no lugar dos mecanismos mais comumente empregados, acarreta em diversas vantagens, como por exemplo:

- **maior rigidez** – esta vantagem está relacionada à distribuição das forças aplicadas aos atuadores devido à disposição paralela, acarretando numa distribuição mais uniforme de esforços;
- **menor inércia** – devido à não serialização dos atuadores e dos módulos cinemáticos, a massa associada aos acionadores é menor, melhorando assim a performance dinâmica do sistema;

- **menor erro de posicionamento e orientação** – esta vantagem ocorre devido ao paralelismo dos atuadores, não acarretando na propagação dos erros de posicionamento de cada atuador ao ser percorrida a cadeia cinemática, vantagem imprescindível em um sistema de posicionamento de precisão;
- **alta repetibilidade** – esta vantagem é intrínseca a um sistema de posicionamento de precisão;
- **altas velocidades de trabalho** – consequência da menor inércia associada ao mecanismo, podendo melhorar a performance do sistema de posicionamento e aumentar o rendimento do sistema;
- **maior simplicidade na solução das equações da cinemática inversa** – esta vantagem está diretamente relacionada à complexidade e tamanho do algoritmo de posicionamento, afetando assim a velocidade de processamento e, consequentemente, de posicionamento;
- **maior complexidade na solução das equações de cinemática direta** – esta desvantagem não afeta diretamente este projeto, pois o posicionamento será feito calculando-se a variação do comprimento dos atuadores para se chegar à posição desejada a partir da inicial (cinemática inversa);
- **menor volume de trabalho** – devido à própria característica do mecanismo por cinemática paralela, o volume de trabalho, isto é, o lugar geométrico dos pontos que podem ser alcançados pelo sistema é menor quando comparado com um sistema serial equivalente.
- **presença de posições singulares** – devido à disposição peculiar dos módulos cinemáticos, exigida para a não interferência entre si, ocorrem posições singulares,

muitas vezes instáveis, limitando severamente as aplicações da Plataforma de Stewart e exigindo rigoroso estudo e adequado sistema de controle.

## **2.2 - Plataforma de Stewart Simplificada**

### **2.2.1 - Histórico**

No mecanismo inicialmente implementado (1998) o posicionamento era obtido através de quatro acionadores lineares, que ligavam a plataforma até a base, através de juntas esféricas.

Em 1999, o projeto foi melhorado e teve a configuração cinemática alterada, dispondo os atuadores lineares fixos, permitindo o decréscimo da inércia da mesa e uma maior estabilidade mecânica do sistema, além de ser exigida menor robustez das juntas e de outras peças. Além disto, esta configuração permitiu a abordagem do sistema de acionamento linear baseado em motores rotativos e de fusos. Entretanto, apesar das possibilidades ampliadas, o acionamento, nesta fase do projeto, era ainda manual (através das manoplas, apresentadas em detalhe).



**Figura 2 - Plataforma de Stewart simplificada em janeiro de 2000.**

Além da definição dos conceitos básicos do projeto e do início da construção do sistema mecânico, os grupos anteriormente orientados pelo Prof. Tarcísio, iniciou os contatos com as empresas SEW e Phoenix Contact, praticamente realizando os contratos de cooperação entre elas e a USP. Estes contatos foram de grande valia, obtendo apoio financeiro e técnico para o desenvolvimento do projeto, sendo obtidos junto à SEW quatro motores trifásicos e quatro conversores de frequência e junto à Phoenix Contact tanto a placa de aquisição e comunicação de dados sob protocolo Interbus como o software de

implementação do sistema. Além disto, apoio técnico e cursos foram prestados por ambas as empresas, proporcionando um abrangente conhecimento sobre o sistema.

A máquina a ser criada para feiras e exposições deveria atender a características de marketing e de apresentação tais que as empresas envolvidas, particularmente a SEW, decidiram por contratar outra empresa, chamada Play Corp, que elaborou o projeto visual da máquina e subcontratou a empresa Droid para a fabricação dos sistemas de suporte e de multimídia.

Coube à USP, através do professor Tarcísio e dos alunos orientados, a elaboração do sistema de posicionamento de precisão por cinemática paralela, incluindo o sistema de controle dos acionadores e dos sensores. Devido à grande abrangência e interconexão entre os sistemas nos módulos de responsabilidade da USP, coube ainda o gerenciamento e integração técnica dos sub-sistemas e das equipes de trabalho envolvidas.

### **2.2.2 - Aplicações possíveis e orientações para o desenvolvimento em paralelo**

Paralelamente à construção de uma máquina para exposições, esteve em andamento o projeto de um centro de usinagem de ferramenta fixa e base móvel, posicionada por esta mesma máquina. Para atender às necessidades deste outro projeto, os meios construtivos e as abordagens tiveram que ser constantemente avaliadas para que não ocorresse um desenvolvimento tal que impossibilitasse o compartilhamento de recursos e de conhecimentos.

Este projeto em andamento (centro de usinagem) é também trabalho orientado pelo Prof. Tarcísio, conduzido pelo aluno Anderson Daigo com previsão de término também para dezembro de 2000.

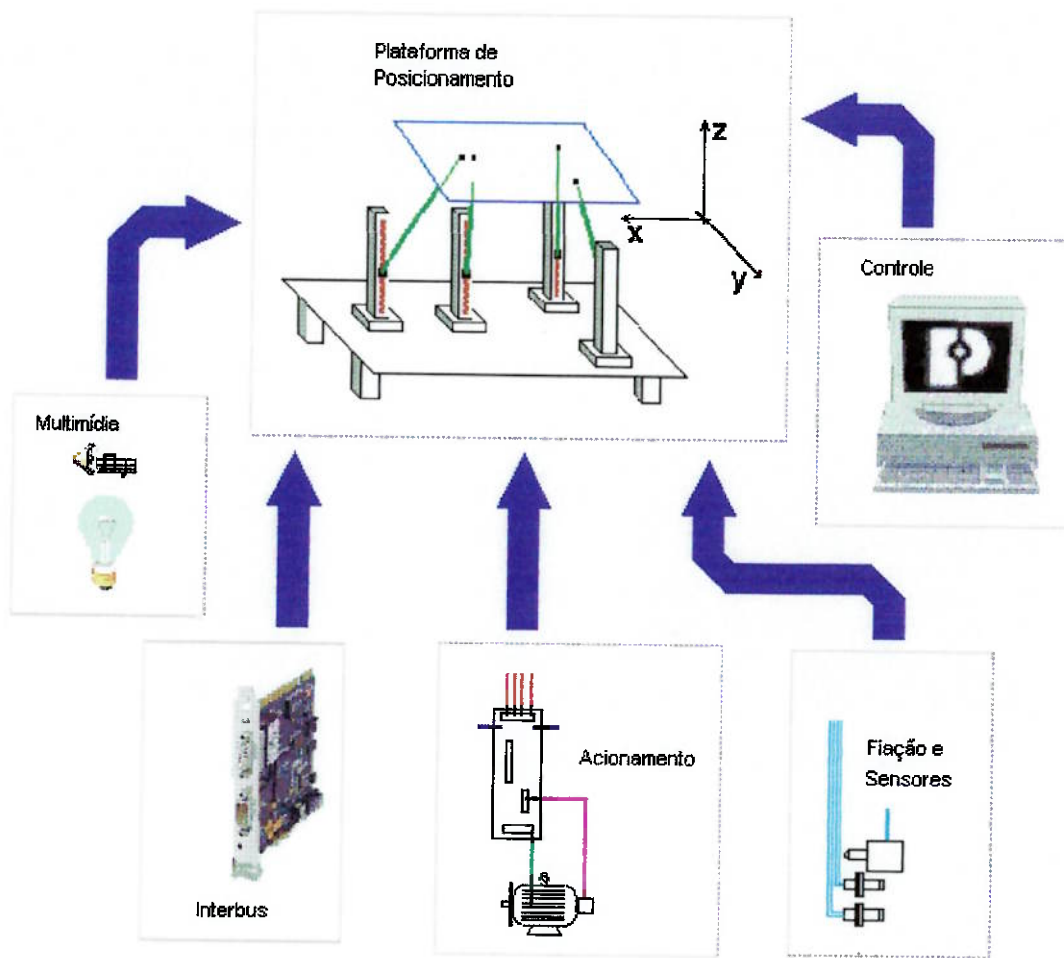
Dado o fato deste trabalho ser um protótipo e devido à limitação de custos envolvidos para a aquisição de partes do sistema, como fusos de precisão e suas respectivas castanhas, fundação para o sistema, entre outros, o rigor necessário para a construção de um sistema de posicionamento de precisão não foi seguido. Desta forma, o protótipo apresenta folgas e deformações que prejudicam seu desempenho, havendo possibilidades aqui para futuras melhorias.

### 3 - Protótipo

Conforme dito anteriormente, o protótipo foi desenvolvido a partir de um mecanismo já projetado, sendo realizadas algumas alterações como redimensionamento das barras e das articulações e modificações drásticas nos sistemas de fixação dos eixos, aumentando sua rigidez. Foram ainda adicionados outros subsistemas, como atuadores controle, o que permite a automatização do posicionamento da plataforma.

Os subsistemas de que se compõe este projeto são:

- subsistema Mecânico
- subsistema de Comunicação e Aquisição de Dados
- subsistema de Acionamento
- subsistema de Controle
- subsistema Elétrico
- subsistema de Multimídia



**Figura 3 - Subsistemas componentes**

Uma descrição simplificada do relacionamento existente entre os subsistemas pode ser vista na Figura 4.

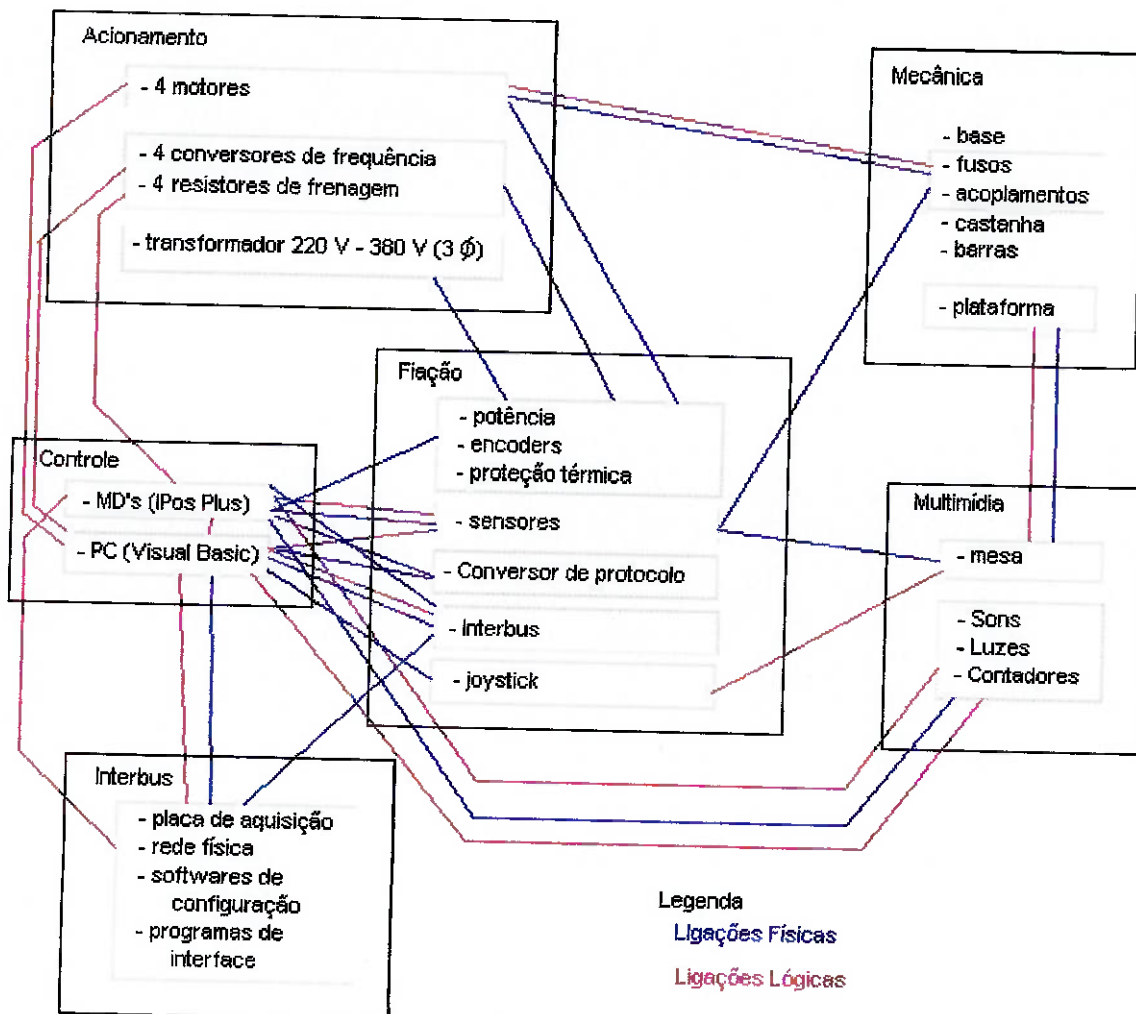


Figura 4 - Inter-relacionamentos entre subsistemas

### 3.1 - Dimensionamento dos Motores

Os componentes do grupo anterior já havia definido alguns parâmetros para a definição do sistema, como a escolha do passo do fuso. Os contatos com a empresa que forneceria os motores (SEW) já tinham razoavelmente definido as características dos

motores e conversores de frequência que seriam utilizados. Entretanto, como a análise das características e o dimensionamento dos motores era inadequado às novas condições, decidiu-se por refazer o dimensionamento dos motores. Este redimensionamento foi feito de acordo com os cálculos apresentados a seguir.

Considerações:

- O rotor do motor possui, juntamente com o fuso e o acoplamento, momento de inércia próximo ao de um disco de  $m_{\text{rotor}} = 1,5 \text{ kg}$ , com raio de  $r = 40 \text{ mm}$ .
- O peso para o qual se propõe o funcionamento do motor é de  $100 \text{ kg}$  (uma mesa com carga total de  $m_{\text{mesa}} = 200 \text{ kg}$ , em configuração tal que o fuso mais solicitado suporte metade desta carga).

Dados:

- Passo do fuso:  $p = 5 \text{ mm}$
- Velocidade média:  $v_{\text{média}} = 0,1 \text{ m/s}$  (para movimento com rampa de aceleração até atingir velocidade máxima seguida de rampa de desaceleração).
- Tempo de rampas de aceleração / desaceleração:  $\Delta t = 0,1 \text{ s}$

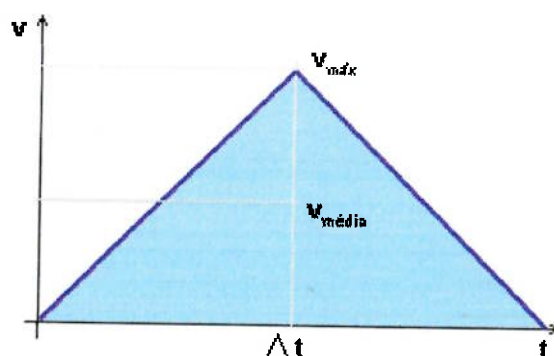


Figura 5 - Movimento adotado como referência para determinação de especificações dos motores

Aceleração da mesa:

$$a = \frac{v_{\max}}{\Delta t} = \frac{2 \cdot v_{\text{média}}}{0,1}$$

$$a = \frac{2,0,1}{0,1} = 2 \text{ m/s}^2$$

Aceleração angular:

$$\alpha_{\text{rotor}} = \frac{2 \cdot \pi \cdot a}{p}$$

$$\alpha_{\text{rotor}} = \frac{2 \cdot \pi \cdot 2}{0,005} = 2512 \text{ rad/s}^2$$

Aceleração angular –  
efeito gravitacional

$$\alpha_{\text{grav}} = \frac{2 \cdot \pi \cdot g}{p}$$

$$\alpha_{\text{grav}} = \frac{2 \cdot \pi \cdot 9,8}{0,005} = 12321,4 \text{ rad/s}^2$$

Momento de Inércia do  
rotor:

$$J_{\text{rotor}} = \frac{m \cdot r^2}{2}$$

$$J_{\text{rotor}} = \frac{1,5 \cdot 0,04^2}{2} = 0,0012 \text{ kg.m}^2$$

Momento de Inércia da  
mesa (abstração):

$$\frac{J_{\text{mesa}}}{2} = \frac{m_{\text{mesa}}}{2} \cdot \left( \frac{p}{2 \cdot \pi} \right)^2$$

$$\frac{J_{\text{mesa}}}{2} = \frac{200}{2} \cdot \left( \frac{0,005}{2 \cdot \pi} \right)^2 = 31,76 \cdot 10^{-6} \text{ kg.m}^2$$

Torque que o motor  
deve ser capaz de  
gerar:

$$\tau_{\text{orque}} = J_{\text{rotor}} \cdot \alpha_{\text{rotor}} + J_{\text{rotor}} \cdot (\alpha_{\text{rotor}} \cdot \alpha_{\text{grav}})$$

$$\begin{aligned} \tau_{\text{orque}} &= 0,0012 \cdot 2512 + \\ &+ 31,76 \cdot 10^{-6} \cdot (2512 + 12321) = \\ &= 3,014 + 0,471 = 3,485 \text{ N.m} \end{aligned}$$

Rotação:

$$\omega_{\max} = \frac{2 \cdot \pi \cdot v_{\max}}{p}$$

$$\omega_{\max} = \frac{2 \cdot \pi \cdot 0,2}{0,005} = 251,33 \text{ rad/s}$$

Potência:

$$Pot_{\max} = \tau_{\text{orque}} \cdot \omega_{\max}$$

$$Pot_{\max} = 3,485 \cdot 251,33 = 875,88 \text{ W}$$

Aplicando-se coeficiente de segurança, chegou-se às seguintes especificações para os motores:

- Potência de saída: 1 kW
- Torque de saída: 5 N.m

Tais especificações condizem com os motores anteriormente concebidos e combinados para fazerem parte do projeto, fazendo com que não fosse necessário alterações nos processos de doação que se encaminhavam.

## **3.2 - Equipamentos doados**

### **3.2.1 - Motores elétricos trifásicos**

Foram doados quatro motores elétricos trifásicos com torque máximo nominal de 10,2 Nm e potência nominal de 1,1 kW. Em condição de uso não-contínuo, estes valores podem chegar a 150% do valor nominal para curtos intervalos de tempo. Como os motores possuem encoders, a malha de controle é semi-fechada, pois não se mede diretamente a posição da mesa.

Os dispositivos mencionados possuem um par de fios ligados por um chapa bi-metálica que proporciona ao equipamento uma proteção térmica quando associado a um conversor com esta função.

Estes motores estão em conformidade com os cálculos de dimensionamento de acionamentos realizados no item 3.1. A Figura 6 apresenta a vista explodida do motor.

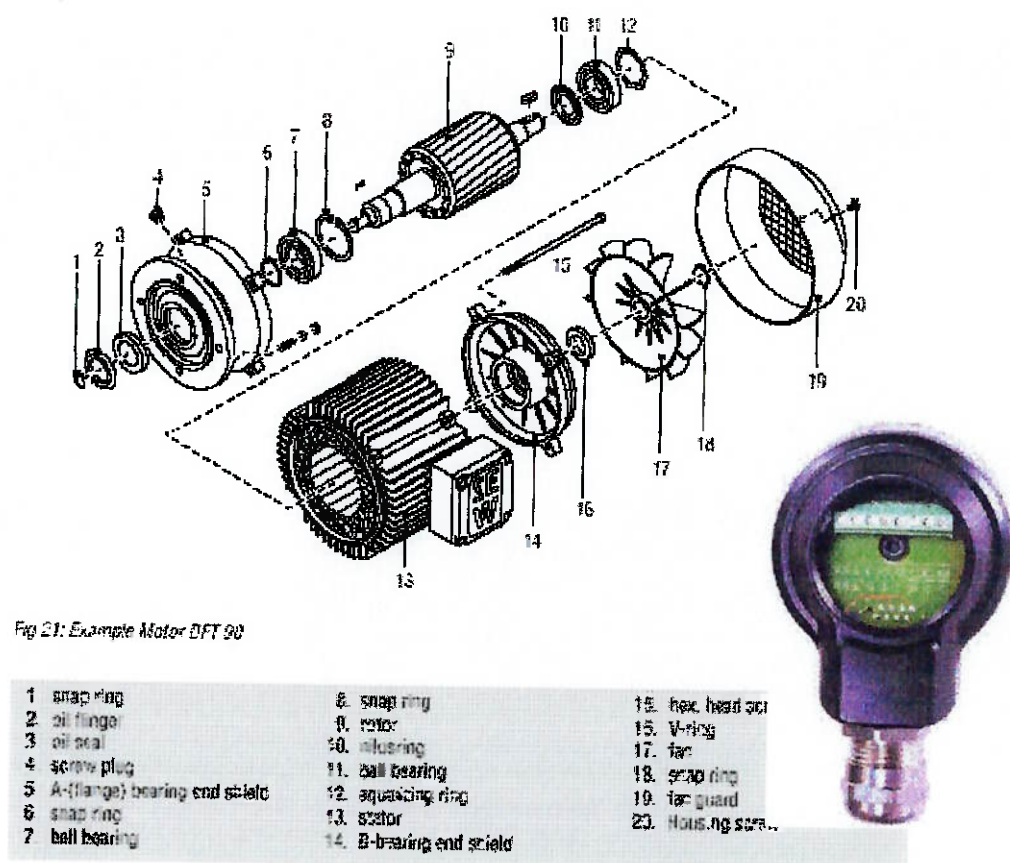
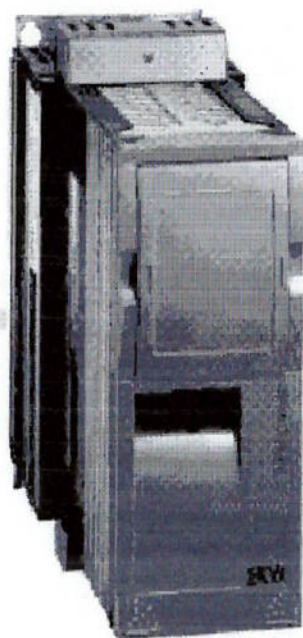


Figura 6 - Vista Explodida do motor com detalhe do encoder.

### 3.2.2 - Conversores de frequência

Os quatro conversores de frequência doados pela SEW são do modelo MOVIDRIVE (MD), modelo MDV 60 A 0015-5A3-4-00, equipamento internamente processado que alimenta e controla motores elétricos trifásicos, sendo capazes de gerenciar sinais de encoders e portanto sendo capazes de fechar a malha de controle dos atuadores. Além disso, através de conectores em seu painel frontal, estes equipamentos possuem entradas e saídas digitais, que através do programa internamente armazenado (em linguagem

específica - IPos Plus), podem ser relacionadas aos “bornes virtuais” (entidades representativas de um estado digital de um bit). Na Figura 7 é mostrada uma foto do conversor de frequência MoviDrive utilizado.



**Figura 7 - Foto de um MoviDrive do modelo indicado.**

Através das placas de comunicação com protocolo Interbus (modelo DFI11A), as informações, tanto para dados de entrada como para dados de saída, podem ser transmitidas para o computador (PC), possibilitando a interação entre chaves, sensores e comunicação do PC com os MoviDrive's, com os sistemas de multimídia e com os operadores externos (pessoal que apresentará o sistema em feiras e eventos). Os bornes virtuais podem ser manipulados e lidos através de palavras de controle e *status*, respectivamente, passadas ou recebidas dos MoviDrive's durante a execução do programa em Visual Basic, existente no PC. O sistema de comunicação implementado através da

placa de aquisição e comunicação de dados e do software relacionado (Interbus) será abordado em maiores detalhes posteriormente.

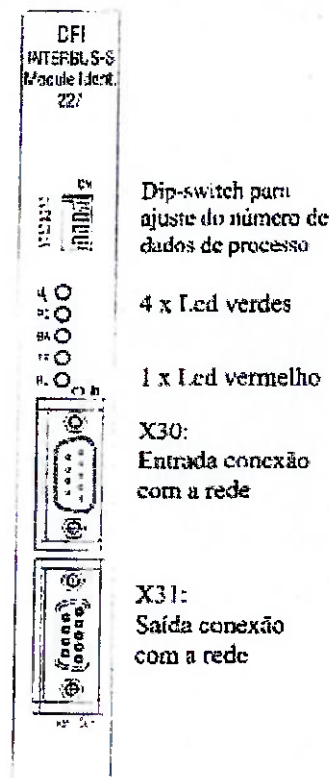


Figura 8 - Placa DFI11A com seus conectores e LED's.

Dados Técnicos do conversor de frequência:

- Alimentação: Trifásica, 380 – 500V, 50 ou 60 Hz;
- Chopper de frenagem integrado;
- 6 entradas binárias isoladas;
- 2 saídas binárias e um relé de saída;
- entradas e saídas programáveis;
- entrada para proteção térmica do motor;
- display para indicação de operação/erro;

- entrada 24V DC externa;
- potência de saída de 2,2 kW para torque variável.

### 3.2.3 - Resistores de Frenagem

Foram doados quatro resistores de frenagem, modelo BW-100-005, pela SEW. Estes resistores, comandados pelos *choppers* de frenagem de seus respectivos MoviDrive's (MD's), foram incorporados ao projeto com o intuito de dissipar uma sobre-tensão gerada pelos motores quando estes trabalham em modo generativo, isto é, quando estão sendo frenados e, em consequência, retornam energia proveniente da energia dinâmica armazenada no sistema.

A característica mais importante que deve ser considerada num resistor de frenagem é a potência regenerativa nominal, levando-se em conta um tempo menor do que 120 s. Para este resistor (100  $\Omega$ ) a faixa de potência regenerativa é de 0,45 kW a 2 kW, dependendo da duração do ciclo de frenagem.

Os resistores de frenagem são diretamente conectados aos MD's. Foram instalados no interior do gabinete e devem receber especial atenção com relação à temperatura de operação, já que estes podem esquentar bastante, em determinadas condições de uso.

### 3.2.4 - Placa de aquisição e comunicação de dados

A placa de aquisição e comunicação doada pela Phoenix Contact é uma placa PCI para PC que realiza a interface entre o PC e o sistema controlado, no caso, os 4 MD's. Na realidade, esta placa envia três palavras (2 bytes cada) de controle para cada um dos MD's e, recebe também três palavras de *status* de cada um dos MD's. Desta forma, pode-se controlar e sensoriar o sistema através destas palavras de controle e de *status*. Nesta comunicação, de acordo com o modo de operação selecionado através do programa armazenado em cada MD (IPos Plus), duas palavras de controle e de *status* comunicam posição e uma palavra de controle e uma de *status* comunicam valores binários, sendo cada bit representativo de um sinal específico (um borne virtual, de escrita ou de leitura, respectivamente). Esta configuração de comunicação deve necessariamente coincidir com a configuração passada para o *Software* de Implementação.

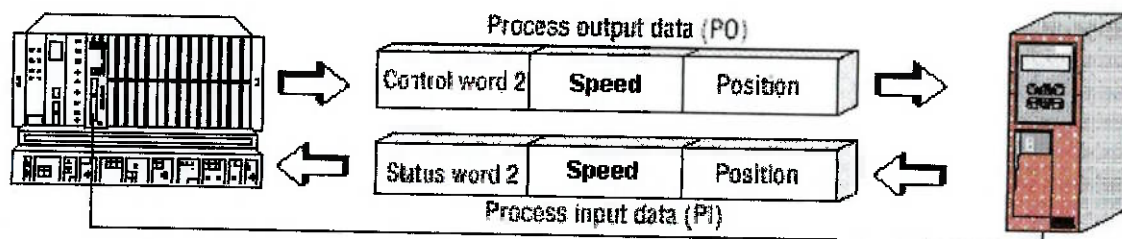


Figura 9 - Exemplo de dados que transitam a via InterBus.

### 3.2.5 - Software de implementação

O software de implementação (CMD), doado pela Phoenix Contact, nada mais é do que um *patch* para compilador, isto é, uma biblioteca de macrofunções pré-desenvolvidas, cuja função é facilitar a implementação da configuração da rede de comunicação Interbus, permitindo um envio e recebimento de palavras de controle e de *status* de maneira mais modular e isento de profundos conhecimentos do funcionamento da placa de aquisição de dados e do sistema Interbus. Este programa consegue determinar o estado da rede física, determinando todos os componentes existentes e sua ordem na rede, (análise de configuração e de erros físicos), auxiliando também na solução de erros de implementação da rede.

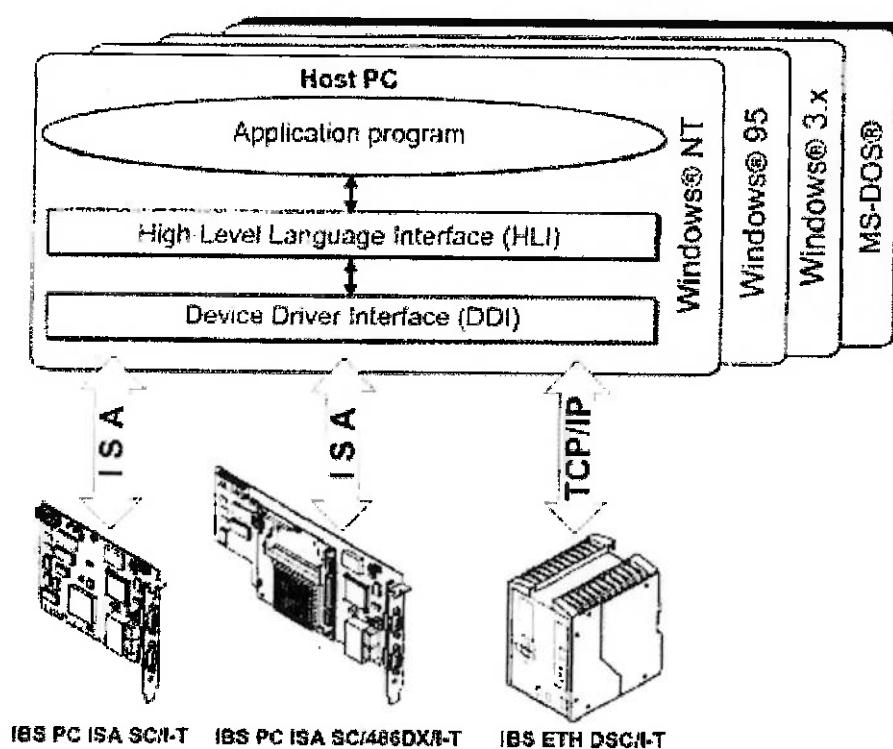


Figura 10 - Estrutura dos drivers de InterBus e interfaces.

### 3.3 - Subsistema mecânico

O subsistema mecânico é composto das seguintes partes:

- **Base** - feita com chapa retangular de aço SAE 1020, de dimensões (1000 X 600 X 10 mm), dotada de pernas reguláveis, para sustentar a estrutura e garantir a ela melhores condições de apoio.
- **Alojamento de acoplamento mecânico** – estrutura construída em alumínio com a finalidade de promover afastamento entre os fusos e os motores, permitindo espaço para a colocação de acoplamento mecânico entre os motores e os fusos. Permite também a desmontagem independente do motor e de seu respectivo fuso.



Figura 11 - Detalhe do alojamento. Os parafusos na sombra permitem retirar o motor sem desmontar a parte superior que contém o fuso.

- **Acoplamento mecânico** – são peças em ferro fundido e borracha que têm por objetivo promoverem um acoplamento macio entre motores e fusos para transmissão de torque.

- **Haste com fuso** –presos ao alojamento do acoplamento e engastados aos acoplamentos. Juntas universais ficam presas pelas castanhas.



Figura 12 - Detalhe do fuso com amortecedor de fim de curso.

- **Barra** – liga a castanha do fuso à plataforma móvel. Construída em aço.
- **Plataforma** –chapa de alumínio para fixação do labirinto ou outro dispositivo. É solidária a 4 juntas universais que transmitem os esforços das barras.

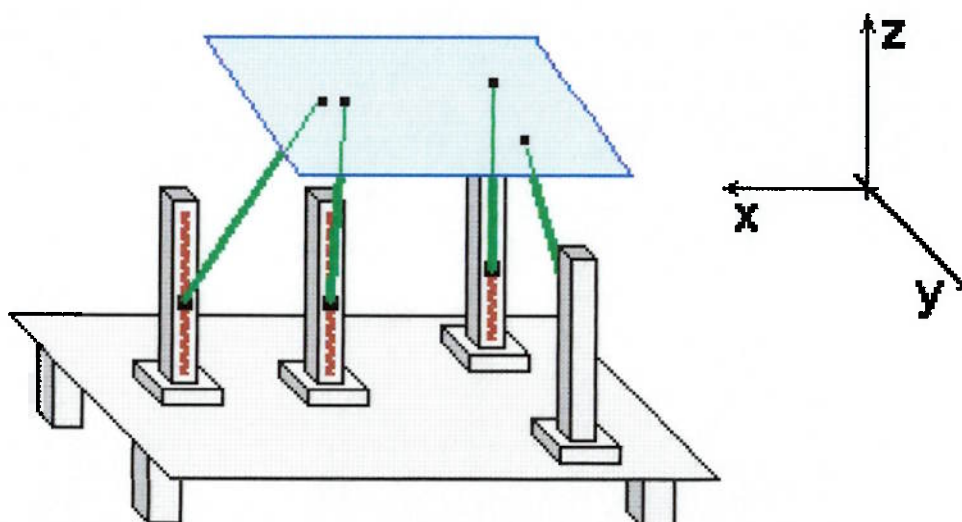


Figura 13 - Figura esquemática dos elementos que compõe o sub-sistema mecânico.

### 3.4 - Subsistema de Comunicação e Aquisição de Dados (Interbus)

O sistema utilizado para aquisição e comunicação de dados é o sistema Interbus, um protocolo de comunicação e aquisição de dados da plataforma FieldBus desenvolvido pela Phoenix Contact. Esta empresa auxiliou no projeto doando a placa de aquisição, ministrando um curso gratuito sobre o sistema de comunicação, seu software e bibliotecas disponíveis.

O CMD (software doado pela Phoenix Contact) configura os recursos do sistema operacional referentes à placa de aquisição (IBS PC ISA SC/I-T) tais como IRQ e intervalo de memória alocado, além de instalar o driver para a placa. Ele possui a opção de ler e identificar os dispositivos presentes na rede Interbus. Desta forma, pode-se modelar o

sistema através de palavras que contém dados de processo (PD's) para cada um dos dispositivos encontrados. A partir disso, pode-se gerar um módulo que contém as declarações das variáveis que representam as PD's e anexar um módulo de linguagem de alto nível (*high level interface* – HLI) que contenha macrofunções de comunicação via Interbus.

O programa gera dois arquivos em Visual Basic (poderia gerar também em outras linguagens como C++ ou Delphi, dependendo de configuração) que podem ser diretamente incorporados como módulos ao projeto do programa em Visual Basic, possuindo três principais funções e variáveis que correspondem aos bits e palavras de controle e de *status* configurados nos MD's (funções e variáveis definidas com escopo global).

Arquivos (módulos ou formulários):

**INTERBUS.BAS** – Implementa as variáveis e funções utilizadas no projeto, tornando-as disponíveis ao programa principal e permitindo o acesso aos endereços correspondentes à placa de aquisição de dados, configurados no arquivo IBSG4HLI.BAS.

**IBSG4HLI.BAS** – Define os endereços lógicos dedicados às funções de comunicação da placa de aquisição de dados e possui funções de comunicação, além de definir tipos de dados e eventos que podem ser úteis para a aplicação.

Funções:

**IBS\_HLI\_Init\_SC1** – Inicializa a comunicação Interbus entre a placa de aquisição de dados e os MD's (executa o ciclo de inicialização da comunicação, inclusive verificando a integridade da rede física Interbus)

**IBS\_HLI\_Process\_SC1** – Processa instruções especificadas (escrita das palavras de controle e leitura das palavras de *status*)

**IBS\_HLI\_Exit\_SC1** – Finaliza comunicação com os MD's

### 3.4.1 - Dados de Processo (*Process Data* – PD's)

Os MoviDrive's podem ser controlados via Interbus através de 1,2 ou 3 palavras contendo os PD's, desde que o número de palavras (*words*) de dados de entrada (PI ou palavras de controle) seja igual ao das palavras de saída (PO ou palavras de *status*). Estas palavras podem ser configuradas de modo a conter dados de:

- Velocidade: com resolução de até 0,2 rpm;
- Corrente: com resolução de até 0,1 % da corrente nominal ( $I_N$ );
- Posição: transmitindo número de pulsos com até 32-bits;
- Velocidade Máxima: configurando o limite de rotação do motor;
- Corrente Máxima: limitando a corrente nas bobinas do motor.

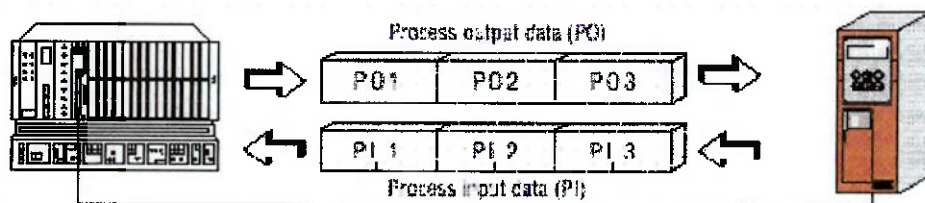


Figura 14 - Exemplo de variáveis de processo (PD's)

No caso específico deste projeto, foram utilizadas *words* de velocidade e posição (nesta ordem), de forma a permitir o controle destes parâmetros. Para tal, faz-se necessária

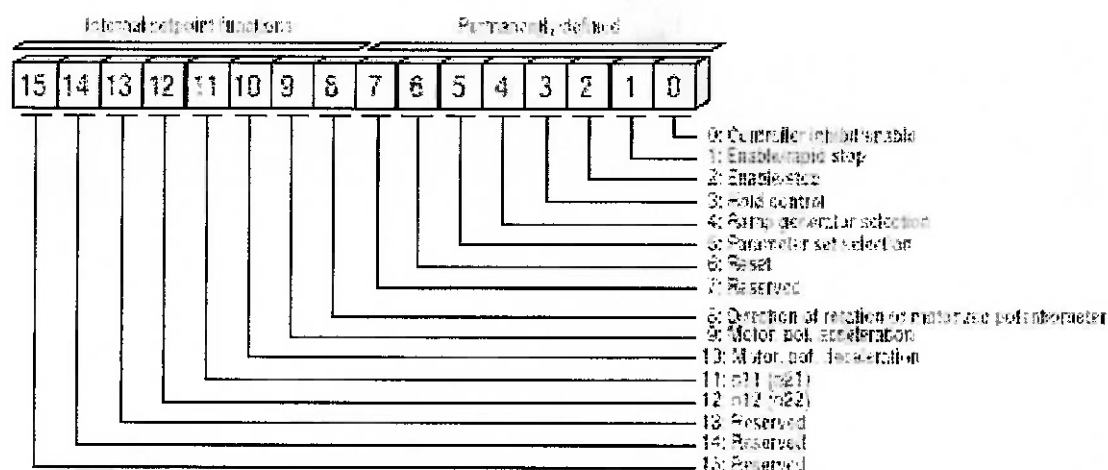
a conversão dos valores a serem transmitidos em hexadecimal. Como cada tipo possui sua conversão a Tabela 1 exemplifica os modos de velocidade e posição.

Tabela 1 - Modos de conversão de velocidade e posição.

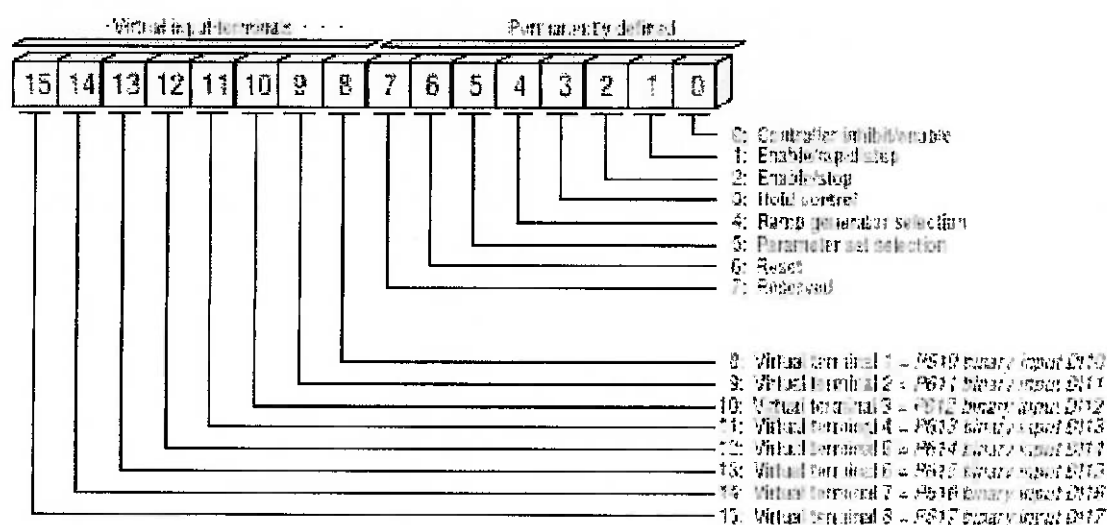
| PD                | Valor                                             | Conversão                        | Conversão Hexadecimal             |
|-------------------|---------------------------------------------------|----------------------------------|-----------------------------------|
| <b>Velocidade</b> | 400 rpm, horário                                  | $\frac{400}{0,2} = 2000_{dec}$   | $2000_{dec} = 07D0_{hex}$         |
|                   | 750 rpm, anti-horário                             | $\frac{-750}{0,2} = -3750_{dec}$ | $-3750_{dec} = F15A_{hex}$        |
| <b>Posição</b>    | 35 voltas, anti-horário,<br>4096 pulsos por volta | $-35.4096 = -143360_{dec}$       | $-143360_{dec} = FFFDD0000_{hex}$ |
|                   | 19 voltas, horário 4096<br>pulsos por volta       | $19.4096 = 77824_{dec}$          | $77824_{dec} = 00013000_{hex}$    |

### 3.4.2 - Palavras de Controle (*Control Words*)

As palavras de controle podem ser do tipo 1 ou do tipo 2, dependendo do tipo de controle que se deseja efetuar. As palavras do tipo 1 contêm funções internas de *setpoint* configuradas bit a bit em seu byte mais significativo (Figura 15a) enquanto que as palavras de controle do tipo 2 (*Control Word 2*) contêm 8 bornes virtuais configuráveis bit a bit (Figura 15b).



(a) Control Word tipo 1.



(b) Control Word tipo 2.

Figura 15 - Tipos de Control Word.

Pode-se notar no entanto que o byte menos significativo (bits de 0 a 7) é idêntico nas duas palavras de controle. Cada bit configura uma função ou ação de controle no respectivo MD, conforme mostra a Tabela 2.

Tabela 2 - Funções bit a bit das Palavras de Controle.

| Bit | Função                                                                     | Opções                                                             |
|-----|----------------------------------------------------------------------------|--------------------------------------------------------------------|
| 0   | Controlador Inibido<br>( <i>Controller Inhibit</i> )                       | 0 = Habilita<br>1 = Controlador Inibido, freio acionado            |
| 1   | Habilita Parada Rápida<br>( <i>Enable/Rapid Stop</i> )                     | 0 = Parada Rápida<br>1 = Habilitado                                |
| 2   | Habilita Parada<br>( <i>Enable/Stop</i> )                                  | 0 = Parar com gerador de rampa<br>1 = Habilitado                   |
| 3   | Manutenção de Posição<br>( <i>Hold Control</i> )                           | 0 = <i>Hold Control</i> Inativo<br>1 = <i>Hold Control</i> Ativado |
| 4   | Seleção de Gerador de Rampa<br>( <i>Ramp Generator Selection</i> )         | 0 = Gerador de Rampa 1<br>1 = Gerador de Rampa 2                   |
| 5   | Seleção de Parâmetro de Configuração<br>( <i>Parameter Set Selection</i> ) | 0 = Parâmetro de Configuração 1<br>1 = Parâmetro de Configuração 2 |
| 6   | Reinício<br>( <i>Reset</i> )                                               | 0 = Inativo<br>1 = Resetar falha                                   |
| 7   | Reservado                                                                  | Deve ser configurado como "0"                                      |

A prioridade entre as funções descritas na Tabela 2 é tanto maior quanto menos significativo for o bit. Isto pode ser visualizado na Figura 16. Sendo assim, não faz sentido configurar o bit referente ao *hold control* (bit 3) se o bit referente a "controlador inibido" (bit 0) for configurado como '1' (controlador inibido).

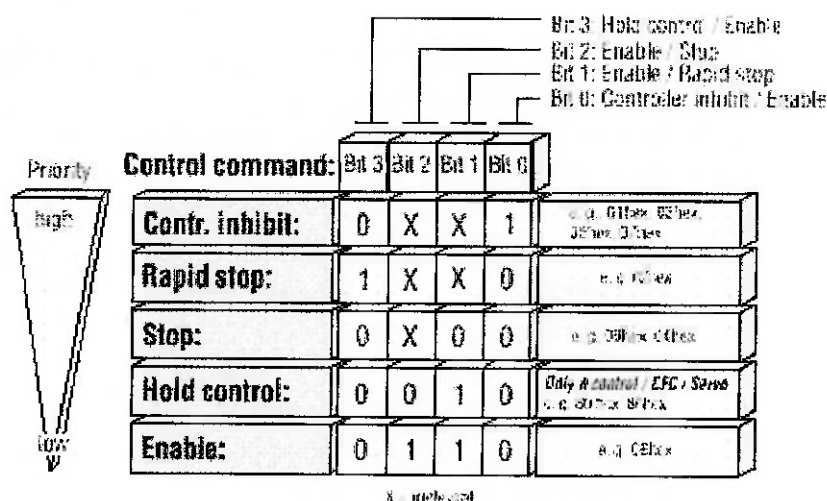


Figura 16 - Prioridade de bits na *Control Word*.

Os itens a seguir descrevem as funções apresentadas na Tabela 2:

- Controlador Inibido – Sobrepõe-se a outros comandos e inibe o funcionamento do motor
- Habilita Parada Rápida – Habilita o rápido travamento do motor
- Habilita Parada – Permite uso de freio mecânico externo
- Manutenção de Posição – É a característica de manter o motor energizado com correntes de 1,5 A para manter sua posição fixa, o que é conseguido por criação de campos magnéticos contraditórios nos enrolamentos do mesmo.
- Seleção de Gerador de Rampa – Seleciona o gerador de rampas de aceleração e desaceleração tipo 1 ou tipo 2, pré-configurados.
- Seleção de Parâmetro de Configuração – Seleciona o parâmetro de configuração tipo 1 ou tipo 2, pré-configurados
- Reset – Libera o MD do estado de travamento após uma falha
- Reservado – Não possui especificação. Por ser reservado, deve ser necessariamente configurado como '0'.

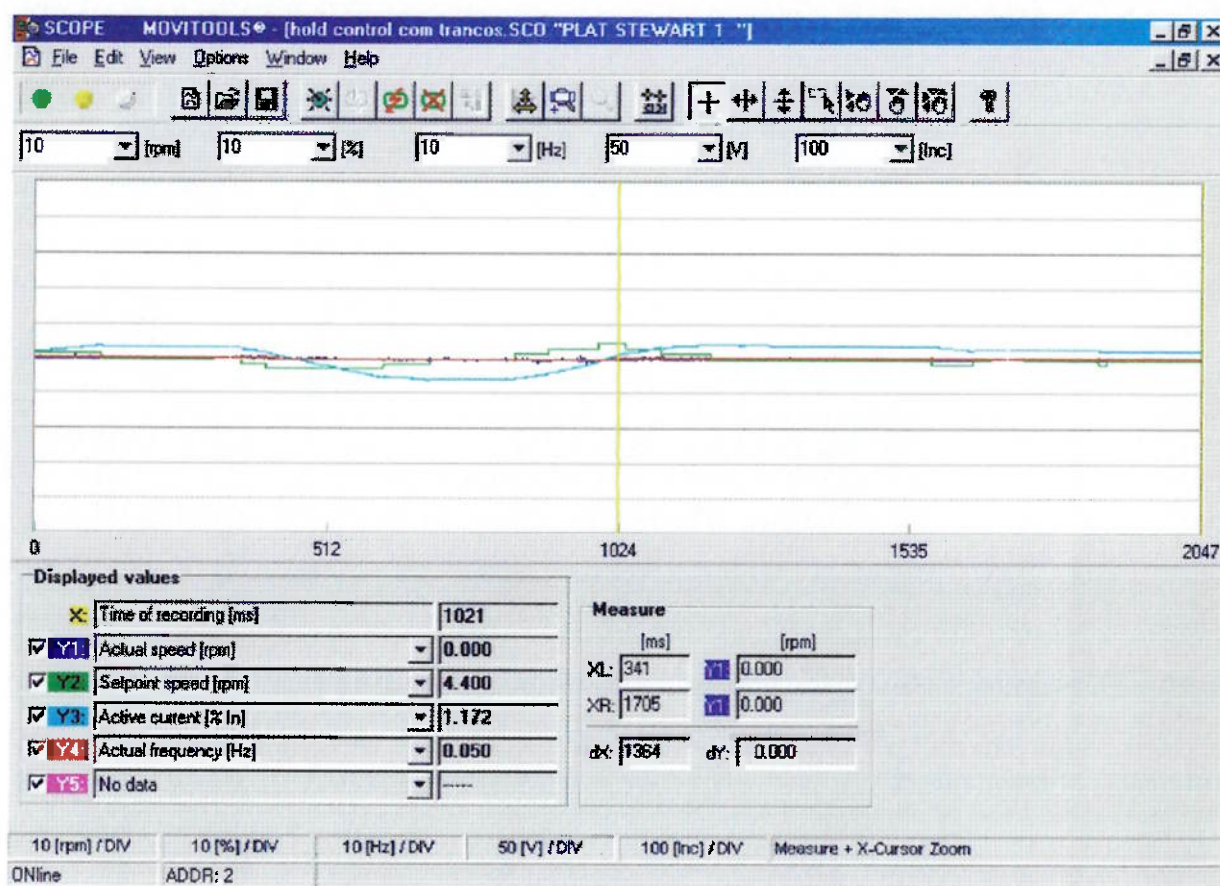


Figura 17 - Gráfico do MD1 em situação de *hold control* sob carga variável.

Este gráfico foi obtido a partir do software MOVITOOLS®, de autoria da SEW, disponível para download pela Internet. A análise do gráfico permite o entendimento do mecanismo de controle (*hold control*) realizado por meio de valores de corrente nos enrolamentos do motor. Pode-se notar que quando a velocidade real (*actual speed* – curva azul escura) difere de zero, o controle gera um *set point* de velocidade não nulo, de forma a tentar estabilizar o motor.

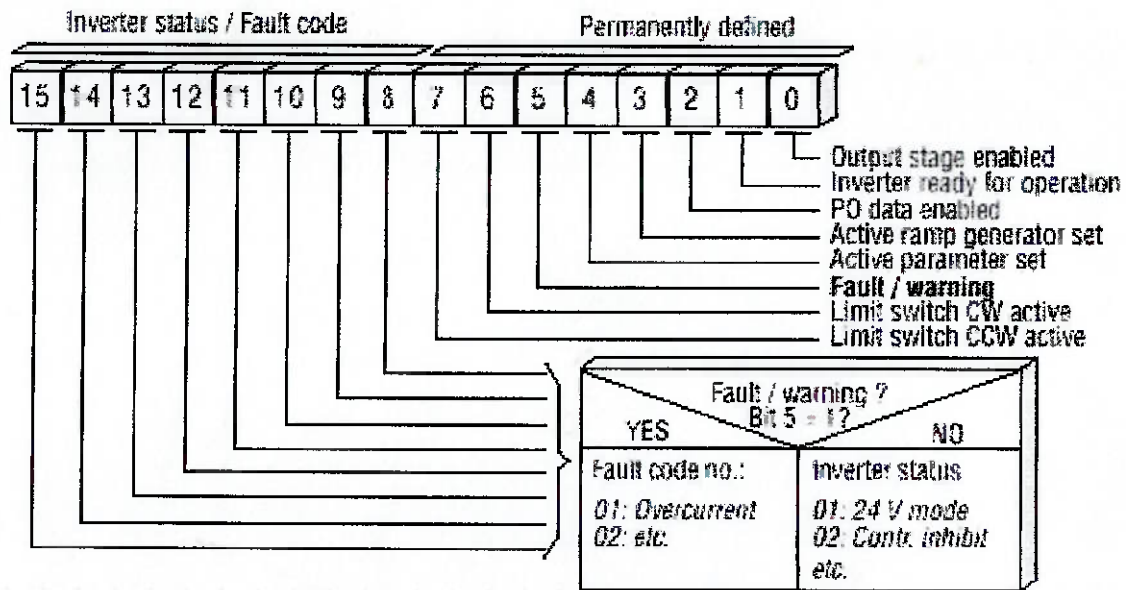
No caso específico deste projeto, utilizou-se Palavras de Controle tipo 2, de forma que o byte mais significativo representa, bit a bit, os 8 bornes virtuais de entrada, isto é, 8 variáveis binárias que são passadas do PC para um determinado MD, de forma a interfacear a lógica de programação de ambos, alterando valores de variáveis no MD. Deve-se salientar neste momento que cada MD possui sua própria *Control Word* e, portanto, cada MD possui 8 variáveis possíveis de entrada e com configurações possivelmente distintas, conforme pode ser visto na Tabela 3.

Tabela 3 - Configuração dos bornes virtuais (bits mais significativos) dos *control words*.

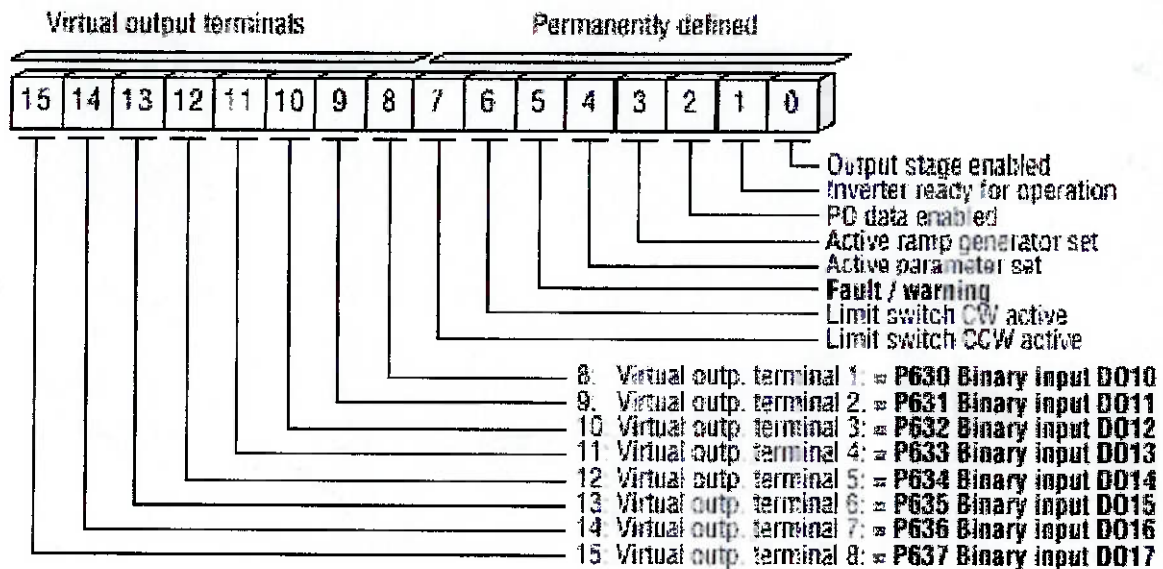
|        | Variável<br>IPos Plus | MD 1            | MD 2 | MD 3 | MD 4 |
|--------|-----------------------|-----------------|------|------|------|
| Bit 8  | DI 10                 | PC pronto       |      |      |      |
| Bit 9  | DI 11                 | Modo Demo ativo |      |      |      |
| Bit 10 | DI 12                 | Modo Jogo ativo |      |      |      |
| Bit 11 | DI 13                 | Jogo iniciado   |      |      |      |
| Bit 12 | DI 14                 |                 |      |      |      |
| Bit 13 | DI 15                 |                 |      |      |      |
| Bit 14 | DI 16                 |                 |      |      |      |
| Bit 15 | DI 17                 |                 |      |      |      |

### 3.4.3 - Palavras de Status

As palavras de *status* assim como as palavras de controle podem ser do tipo 1 ou do tipo 2. As palavras do tipo 1 contêm informações sobre ocorrência de falhas ou avisos e quais foram eles nos bits mais significativos (Figura 18a) enquanto que as palavras de controle do tipo 2 (*Control Word 2*) contêm 8 bornes virtuais configuráveis bit a bit conforme Figura 18b.



(a) Status Word tipo 1.



(b) Status Word tipo 2.

Figura 18 - Tipos de Status Words.

Pode-se notar que, assim como nas palavras de controle, o byte menos significativo (bits de 0 a 7) é idêntico nas duas palavras de *status*. Cada bit configura um estado no respectivo MD, conforme mostra a Tabela 4.

Tabela 4 - Estados bit a bit das Palavras de *Status*.

| Bit | Função                                                                      | Opções                                                                 |
|-----|-----------------------------------------------------------------------------|------------------------------------------------------------------------|
| 0   | Estágio de Saída Habilitado<br>( <i>Output Stage Enabled</i> )              | 0 = Desabilitado<br>1 = Habilitado                                     |
| 1   | MD Pronto para Operação<br>( <i>Inverter Ready for Operation</i> )          | 0 = MD Não Pronto<br>1 = MD Pronto                                     |
| 2   | Dados de Saída de Processo Habilitados<br>( <i>PO Data Enabled</i> )        | 0 = Dados Desabilitados<br>1 = Dados Habilitados                       |
| 3   | Gerador de Rampa Ativa Configurado<br>( <i>Active Ramp Generator Set</i> )  | 0 = Gerador Não Configurado<br>1 = Gerador Configurado                 |
| 4   | Parâmetro de Configuração Ativo Conf.<br>( <i>Active Parameter Set</i> )    | 0 = Parâmetro de Conf. Não Conf.<br>1 = Parâmetro de Conf. Configurado |
| 5   | Ocorrência de Falha ou Aviso<br>( <i>Fault/Warning</i> )                    | 0 = Parâmetro de Configuração 1<br>1 = Parâmetro de Configuração 2     |
| 6   | Chave de Fim de Curso Horária Ativada<br>( <i>CW Limit Switch Active</i> )  | 0 = Chave Fim de Curso Desativada<br>1 = Chave Fim de Curso Ativada    |
| 7   | Ch. Fim de Curso Anti-Horária Ativada<br>( <i>CCW Limit Switch Active</i> ) | 0 = Chave Fim de Curso Desativada<br>1 = Chave Fim de Curso Ativada    |

Os itens a seguir descrevem os estados apresentadas na Tabela 4:

- Estágio de Saída Habilitado – estágio de saída de potência do conversor habilitado
- MD Pronto para Operação – MD está pronto para iniciar operação, isto é, seu *start-up* foi feito e sua *checklist* verificada

- Dados de Saída de Processo Habilitados – Habilita os dados de saída do MD, isto é, os próximos bits possuem valores atribuídos.
- Gerador de Rampa Ativa Configurada – O gerador de rampa ativa (1 ou 2) foi ativado
- Parâmetro de Configuração Ativo Configurado – O parâmetro de configuração ativo (1 ou 2) foi ativado
- Ocorrência de Falha ou Aviso – ocorrência de falha ou aviso conforme listado em Anexo A.
- Chave de Fim de Curso Horária e Anti-Horária Ativada – em alguns projetos, de forma a proteger a planta, configura-se por *software* (IPos Plus) limites extremos para rotação do motor. Estes limites são configurados em termos de pulsos em relação à referência e geram uma falha quando ativados.

No caso específico deste projeto, utilizou-se Palavras de *Status* tipo 2, de forma que o byte mais significativo representa, bit a bit, os 8 bornes virtuais de saída, isto é, 8 variáveis binárias que são passadas do MD para o PC, de forma a interfacear a lógica de programação de ambos, alterando valores de variáveis no PC. Deve-se salientar neste momento que cada MD possui sua própria *Status Word* e, portanto, cada MD possui 8 variáveis possíveis de saída e com configurações possivelmente distintas, conforme pode ser visto na Tabela 5.

Tabela 5 - Configuração dos bornes virtuais (bits mais significativos) dos *status words*.

|        | Variável<br>IPos Plus | MD 1                 | MD 2      | MD 3 | MD 4 |
|--------|-----------------------|----------------------|-----------|------|------|
| Bit 8  | DO 10                 | MD pronto            |           |      |      |
| Bit 9  | DO 11                 | Motor referenciado   |           |      |      |
| Bit 10 | DO 12                 | Movimento finalizado |           |      |      |
| Bit 11 | DO 13                 | Sensor SP            | Sensor CB | cDH  |      |
| Bit 12 | DO 14                 | Sensor SG            | Sensor BP | cDF  |      |
| Bit 13 | DO 15                 | CTE                  |           |      |      |
| Bit 14 | DO 16                 |                      |           |      |      |
| Bit 15 | DO 17                 |                      |           |      |      |

#### 3.4.4 - Variáveis Lógicas de Programação

Como mencionado anteriormente, cada MD possui suas respectivas palavras de controle e *status*. Além disso cada ordem de controle ou de *status* possui 3 *words*, a primeira de configuração, a segunda de velocidade e a última de posição. Desta forma, deve-se configurar 24 variáveis correspondentes a essas palavras conforme Tabela 6.

Tabela 6 - Variáveis Lógicas de Programação.

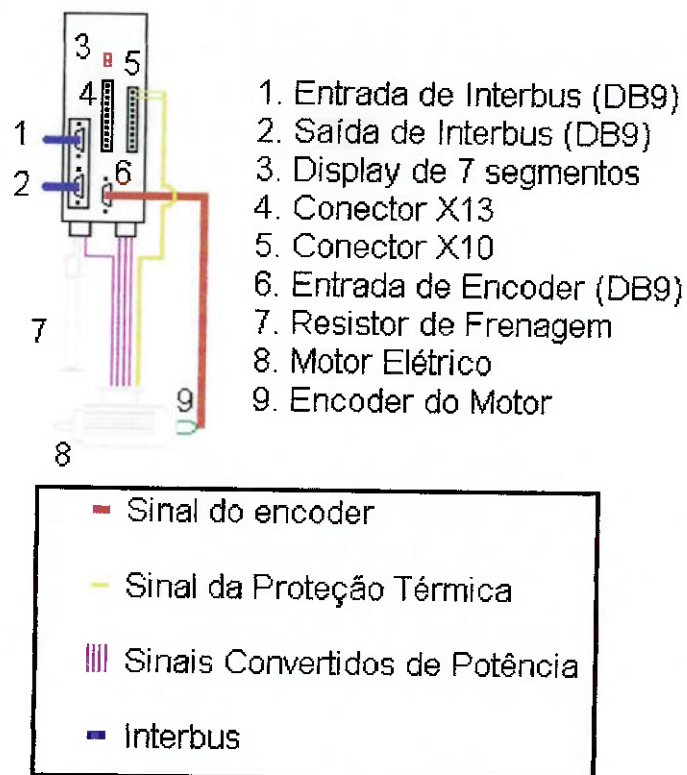
| Variável                                        | Referência                                                          |
|-------------------------------------------------|---------------------------------------------------------------------|
| PLAT_1_STATUS<br>PLAT_2_STATUS<br>PLAT_3_STATUS | Leitura do estado dos bornes virtuais de<br>leitura dos quatro MD's |

|                                                                      |                                                                                              |
|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| PLAT_4_STATUS                                                        |                                                                                              |
| PLAT_1_POS_STA<br>PLAT_2_POS_STA<br>PLAT_3_POS_STA<br>PLAT_4_POS_STA | Leitura da posição dos encoders dos quatro MD's (sinais vindos dos motores)                  |
| PLAT_1_VEL_STA<br>PLAT_2_VEL_STA<br>PLAT_3_VEL_STA<br>PLAT_4_VEL_STA | Leitura do estado de velocidades dos quatro MD's                                             |
| PLAT_1_CONTROL<br>PLAT_2_CONTROL<br>PLAT_3_CONTROL<br>PLAT_4_CONTROL | Alteração de estado dos bornes virtuais de escrita dos quatro MD's                           |
| PLAT_1_POS<br>PLAT_2_POS<br>PLAT_3_POS<br>PLAT_4_POS                 | Ordem de deslocamento até as posições de encoders especificadas para cada um dos quatro MD's |
| PLAT_1_VEL<br>PLAT_2_VEL<br>PLAT_3_VEL<br>PLAT_4_VEL                 | Ordem de deslocamento com velocidades (em módulo) especificadas para cada um dos quatro MD's |

Estas variáveis são implementadas pelo *patch* da Phoenix Contact sob a forma de 2 módulos a serem anexados ao projeto (em Visual Basic ou C++) como mencionado anteriormente.

### 3.5 - Subsistema de Acionamento

O subsistema de acionamento é composto por 4 motores trifásicos, de potência 1,5 kW, alimentados por conversores de frequência (MoviDrive's) doados pela empresa SEW do Brasil. Os MoviDrive's implementam os sinais de potência de maneira que os motores sigam comandos gerados pelo PC, passados por meio do sistema de comunicação (Interbus). Nestes comandos estão determinados a posição final e a velocidade máxima esperada para o deslocamento de cada motor. Além de gerarem os sinais de potência, os MoviDrive's monitoram os motores (através de encoders de 4096 pulsos / volta, existentes nos motores) e também servem como conexão com sensores e chaves, transformando o sinal elétrico proveniente destes em dados que serão processados pelos próprios MD's como pelo computador via Interbus. Desta forma, pode-se enviar e receber palavras de controle pelo computador, gerando assim o fluxo de comandos através de um sistema de controle.



**Figura 19 - Unidade de Acionamento**

Os MoviDrives foram configurados em modo CFC (controle de fluxo por corrente), isto é, o motor é comandado por corrente, o que confere a característica de um acionamento com torque controlado, ao invés de VFC (controle de fluxo por tensão) onde a velocidade do rotor é controlada.

Os modos acima descritos podem ser comparados através da Tabela 7:

**Tabela 7 - Comparação entre modos VFC e CFC**

| VFC – Controle de fluxo por tensão   | CFC – Controle de fluxo por corrente |
|--------------------------------------|--------------------------------------|
| Torque máximo de 150% do nominal     | Torque máximo de 160% do nominal     |
| Características similares a um servo | Características de servo             |

|                         |                                    |
|-------------------------|------------------------------------|
| Uso de encoder opcional | Uso de encoder obrigatório         |
| Maior tempo de resposta | Menor tempo de resposta            |
|                         | Controle de torque                 |
|                         | Alta precisão e dinâmica de parada |

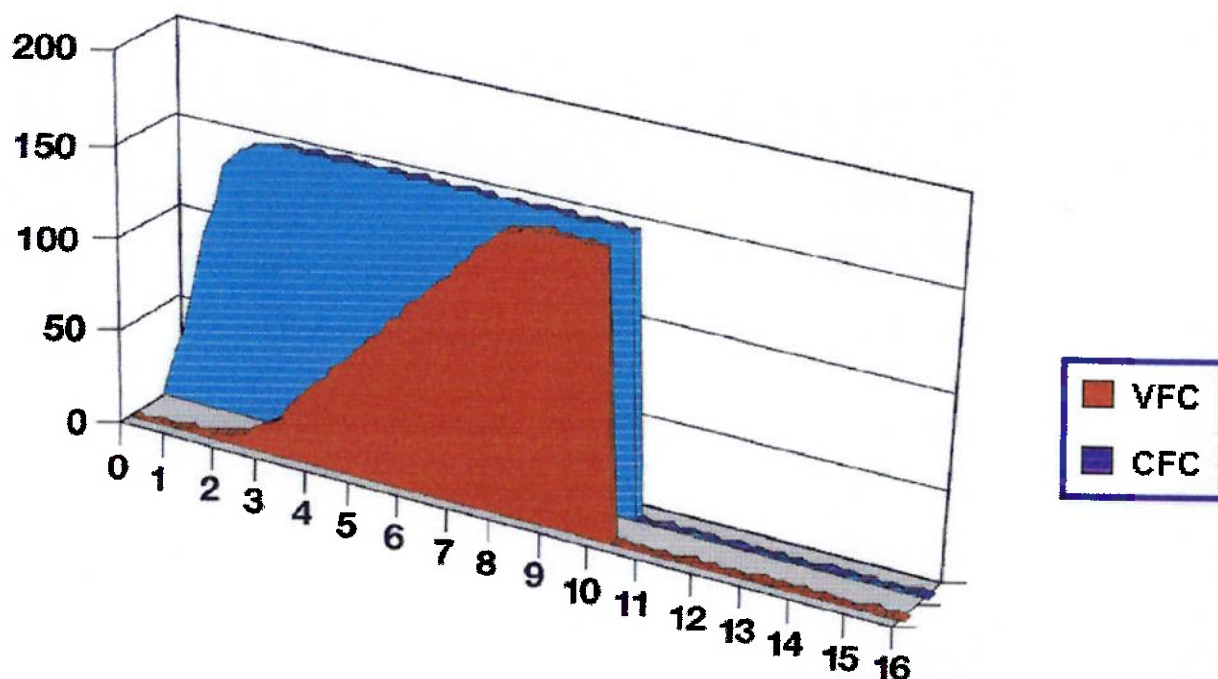
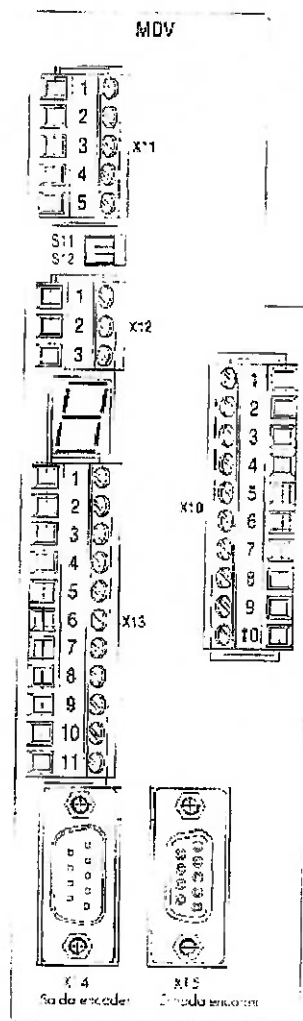


Figura 20 - Gráfico de comparação do tempo de resposta dos modos VFC e CFC.

### 3.5.1 - Painel Frontal do MD

A disposição de conectores e de outros elementos do painel frontal pode ser visualizada na Figura 21.



**Figura 21 - Painel frontal do MD.**

As tabelas 8, 9 e 10 apresentam as principais funcionalidades e interfaces implementadas no painel frontal.

O único elemento de interface com o usuário presente no painel é o display de 7 segmentos, o qual indica a condição de operação ou condição de falha. O código de operação é apresentada na Tabela 8 e o código de falha no Anexo A.

**Tabela 8 - Opções de operação conforme *display* de 7 segmentos**

|   |                                   |   |                                  |
|---|-----------------------------------|---|----------------------------------|
| 0 | Conversor não-pronto              | 9 | Limite de fim de curso alcançado |
| 1 | Controlador inibido               | A | Opção de tecnologia              |
| 2 | Sem permissão (*)                 | b | Livre                            |
| 3 | Corrente de manutenção de posição | c | Modo de referenciamento por IPos |
| 4 | Operação em modo VFC              | d | Reinicialização sem parada       |
| 5 | Controle de velocidade            | E | Livre                            |
| 6 | Controle de torque                | F | Indicação de falha (piscando)    |
| 7 | "Hold control"                    | H | Operação manual                  |
| 8 | Configuração de fábrica           | t | Timeout ativo                    |

(\*) Neste projeto, os displays de 7 segmentos de todos os conversores de frequência devem estar em 2 para início de comunicação via Interbus.

No painel frontal encontram-se diversos conectores. São os seguintes:

- X10 – Saídas Digitais Chaveadas e alimentação externa 24 V DC.
- X11 – Entradas Analógicas
- X12 – Barramento de Sistema
- X13 – Entradas Digitais
- X14 – Saída do Encoder (DB9 macho)
- X15 – Entrada do Encoder (DB9 fêmea)

Destes conectores, destacam-se pela sua intensa utilização no projeto os conectores X10, X13 e X15.

As funções presentes no conector X10 são apresentados na Tabela 9:

**Tabela 9 - Conector X10.**

| Número do pino | Identificação | Função                     |
|----------------|---------------|----------------------------|
| 1              | TF1           | Entrada de TF              |
| 2              | DGND          | Terra para sinais binários |
| 3              | DB00          | Freio                      |
| 4              | DO01-C        | Contato comum do Relé      |
| 5              | DO01-NO       | Contato NA do Relé         |
| 6              | DO01-NC       | Contato NF do Relé         |
| 7              | DO02          | Falha                      |
| 8              | VO24          | + 24 V                     |
| 9              | VI24          | + 24 V de fonte externa    |
| 10             | DGND          | Terra para sinais binários |

Os conectores X11 e X12, bem como os cabos que nele se acoplam estão descritos no Anexo D, referente à fiação.

Os bornes virtuais e outros sinais presentes no conector X13 são apresentados na Tabela 10:

**Tabela 10 - Conector X13.**

| Número do pino | Identificação | Função |
|----------------|---------------|--------|
|----------------|---------------|--------|

|    |      |                  |
|----|------|------------------|
| 1  | DI00 | Controle inibido |
| 2  | DI01 | Borne virtual 1  |
| 3  | DI02 | Borne virtual 2  |
| 4  | DI03 | Borne virtual 3  |
| 5  | DI04 | Borne virtual 4  |
| 6  | DI05 | Borne virtual 5  |
| 7  | DCOM | Terra            |
| 8  | VO24 | Saída +24 V      |
| 9  | DGND | Terra            |
| 10 | ST11 | RS-485(+)        |
| 11 | ST12 | RS-485(-)        |

O conector X14 não é utilizado para este projeto.

O conector X15 é descrito no Anexo D.

### 3.5.2 - Placas de comunicação para protocolo Interbus (DFI11A)

A placa de comunicação para implementação de protocolo Interbus no MD (todos os quatro MD's estão equipados com uma) é o elemento de interface entre o MD e a via de dados (Interbus). Ela é responsável pela decodificação dos sinais para a interface com o MD. Nela estão presentes dois conectores para acoplamento da rede física Interbus (cabos), sendo eles:

- DB9-macho – Entrada Interbus

- DB9-fêmea – Saída Interbus

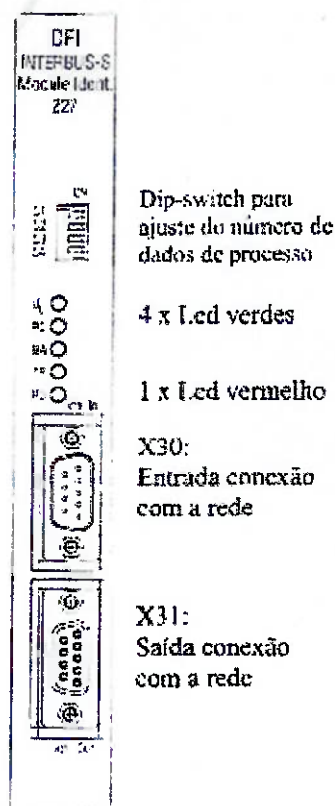


Figura 22 - Placa de Interface MD – Interbus

Possui ainda quatro LED's que têm por finalidade indicar ao usuário o estado da rede Interbus. Estes LED's são padronizados para protocolo FieldBus. São eles:

- UL – Alimentação está OK
- RC – Conexões físicas (cabos) OK
- BA – Indica o ciclo que está sendo executado (aceso = ciclo de dados)
- RD – Elemento da rede desabilitado por software
- ERR – Erro
- US ou UB – Alimentação auxiliar OK

Uma explicação mais detalhada sobre protocolo de controle e sensoreamento FieldBus é apresentada no Anexo G.

### 3.6 - Subsistema de Controle

O subsistema de controle consiste de 2 níveis: o primeiro é implementado pelo microcomputador e auxiliado pelo sistema de comunicação e aquisição de dados Interbus. Nele se encontra em operação um programa, construído em Visual Basic, que controla as operações de alto nível (o código-fonte deste programa pode ser encontrado no Anexo B); o segundo sistema de controle é implementado em cada conversor de frequência (MoviDrive), que opera conjuntamente com sinais de informação vindos do computador e com sensores e atuadores (chaves, motores, encoders, sensores de posição da bola, sensores de fim de curso, etc). Ele gerencia todas as informações e gera sinais de potência para acionar adequadamente os motores trifásicos. Através de informações vindas dos encoders, após processadas, os MD's disponibilizam, através das palavras de *status* do protocolo Interbus, informações sobre posições e velocidades dos motores. O código-fonte dos programas dos MD's pode ser encontrados no Anexo C.

### 3.7 - Subsistema elétrico

É composto por vários conjuntos de fiação. Esta fiação tem a função de ligar os sensores, motores e chaves ao sistema de controle. O cabo que estabelece a rede física do Interbus pertence ao subsistema de comunicação mas, por didática, pode ser abordado também como pertencente ao subsistema elétrico.

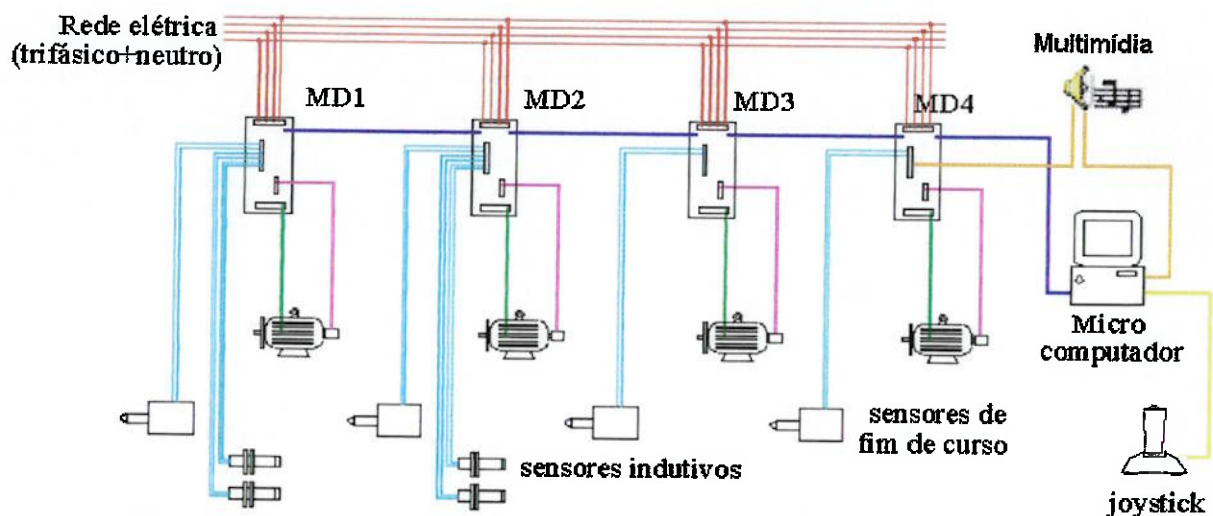
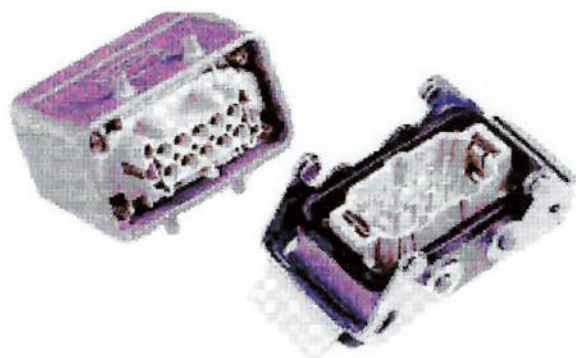


Figura 23 - Esquema elétrico de integração dos diversos sub-sistemas.

As principais partes da fiação são as seguintes:

- Potência – engloba a ligação do equipamento à rede elétrica (trifásica, 220 V), o transformador trifásico (220 V – 380 V), a ligação deste ao equipamento e a interligação de potência dos MD's. Também é considerada fiação de potência a fiação que alimenta os motores a partir das saídas de potência dos MD's e a ligação com os resistores de frenagem.



**Figura 24 - Conector de potência utilizado para ligar o gabinete à mesa.**

- Interbus – estabelece a comunicação entre a placa de aquisição de dados com os MD's. Compõe-se de um único cabo entre cada MD e o MD vizinho e um cabo de um dos MD's (MD4) até o computador.
- Encoders – a fiação de encoders leva o sinal dos encoders até o MD que controla o respectivo motor. Compõe-se a parte de fiação de encoders apenas os quatro cabos que ligam os motores (onde estão os encoders) aos MD's.
- Proteção Térmica – os motores são dotados de sensores térmicos. Cada sensor gera um sinal que pode ser interpretado pelo MD que o controla e assim, em caso de sobreaquecimento do motor, o MD suspende seu funcionamento. É implementada por um cabo desde a base da mesa (onde estão ligados aos sensores térmicos dos motores) ao gabinete e deste aos respectivos MD's.
- Sensores – esta parte da fiação liga os sensores (sensores de fim de curso e sensores de mesa) aos MD's. Esta ligação física é correspondente a uma ligação lógica existente no programa armazenado no MD, devendo ser respeitada. O cabeamento vem desde a mesa, junta-se com os sensores de fim de curso e daí até o gabinete. Do gabinete distribui-se para os quatro MD's.

- Conversor RS485 para RS232 – esta parte da fiação é responsável pela ligação da área de armazenamento de programas dos MD's com o meio exterior, sendo por esta feita a programação e supervisão do funcionamento interno dos MD's (programa MOVITOOLS®). É utilizada desde o MD1 até um conversor de protocolo (RS232 – RS485) e daí até o gabinete e deste até o computador de programação dos MD's. Existe também a comunicação entre os MD's, através de dois fios, que são encaminhados através dos cabos utilizados para a implementação de outras partes da fiação. Uma vez validado o funcionamento do equipamento, a conexão entre o gabinete e computador que programa os MD's não precisa mais ser utilizada.

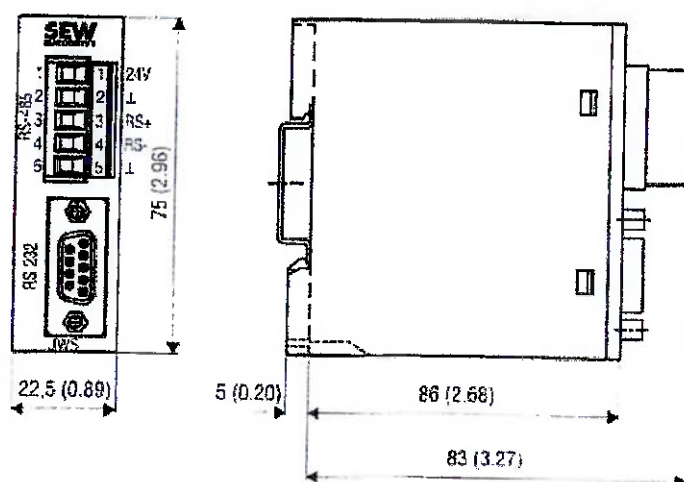


Figura 25 - Vistas do conversor de protocolo RS485 para RS232.

- Chaves e Sinais – esta parte da fiação, que é implementada no interior do gabinete, é responsável pela comunicação entre os sistemas de mídia (exceto som) com os MD's.
- Joystick – compõe-se de uma extensão do cabo do joystick, ligando-o até o gabinete e deste com o PC.

No Anexo D encontram-se diversas tabelas que apresentam as interconexões que implementam as interfaces entre os sub-sistemas descritos e a pinagem e coloração dos fios utilizados na fiação.

## 4 - Máquina de feira

Conforme mencionado anteriormente, um dos objetivos deste projeto é a fabricação de uma máquina a ser apresentada em feiras, congressos e convenções, de forma a expor os equipamentos doados, em especial os conversores de frequência, MoviDrive's.



Figura 26 - Foto da máquina de feira.

Foram planejados dois modos de operação do equipamento quando em situação de exposição: modo demo e modo jogo.

Sobre a plataforma é presa uma mesa, onde é aplicado o tabuleiro do jogo. Esta mesa possui quatro sensores indutivos, responsáveis pela leitura da posição atual da esfera metálica que faz parte do jogo. Esta mesa tem também a função de tornar o sistema mais amigável, visualmente.

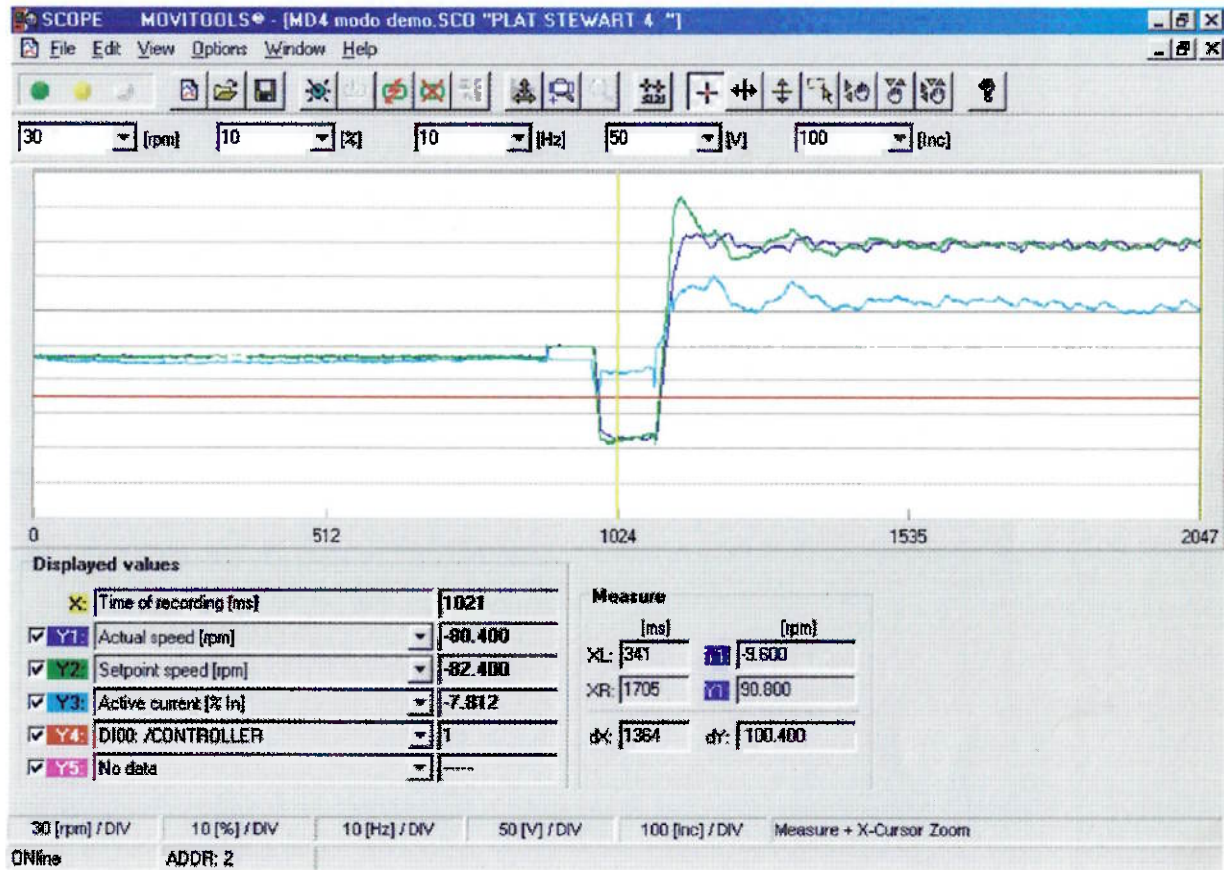
A mesa pode ser substituída, já que esta foi feita separadamente, aumentando a modularidade do sistema, permitindo alterações no labirinto ou no sistema de jogo sem que sejam necessárias maiores modificações no restante do sistema.

## **4.1 - Modo demo (demonstrativo)**

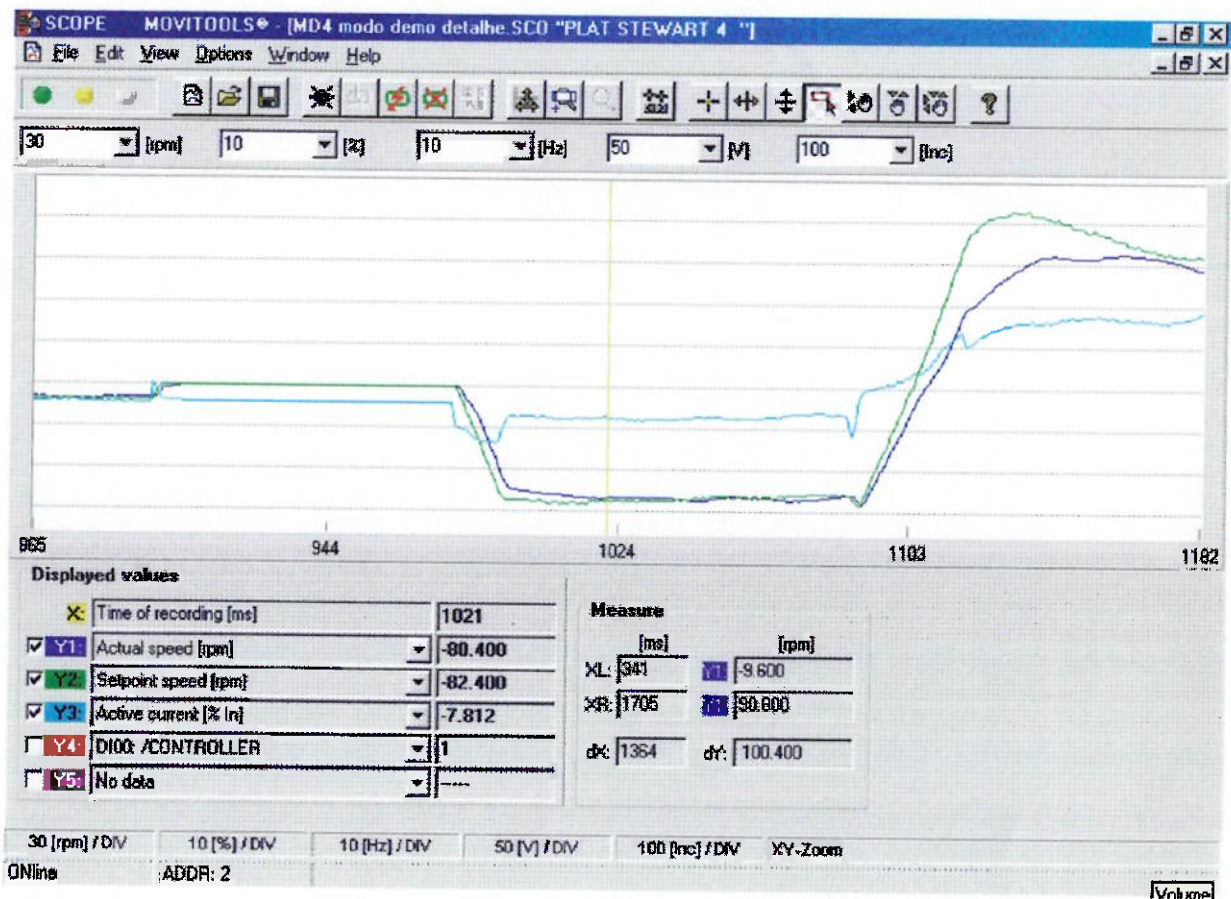
Neste modo, o programa do PC gera 4 números aleatórios compreendidos em intervalos de valores especificados. Estes números correspondem à posição do encoder que se pretende que cada um dos quatro motores alcancem. Após a finalização do movimento (isto é, após todos os encoders acusarem um número de pulsos igual ao respectivo número de pulsos que foi passado), outros 4 números aleatórios são gerados e passados aos motores de forma que a plataforma fica, continuamente, movendo-se, seguindo posições definidas aleatoriamente.

Há também, como requisitado pela SEW, um botão na parte traseira do gabinete que habilita ou desabilita o modo demo, isto é, quando este botão (cHD) é acionado, a mesa fica horizontalmente nivelada e estática, sem que sejam gerados valores aleatórios para a sua movimentação. Além disso, o som, que é passado pela placa multimídia do PC para o amplificador de som, fica desabilitado. Quando o modo demo está habilitado, um outro

gerador de números aleatórios escolhe uma dentre 4 músicas pré-determinadas a ser tocada durante a apresentação e a geração de valores aleatórios para posição de encoder voltam a ser gerados.



(a) Gráfico do MD4 durante modo demo com inversão de sentido.



(b) Detalhe da inversão de sentido do MD4.

Figura 27 - Gráfico do MD4 durante modo demo.

## 4.2 - Modo jogo

Para o modo jogo, sobre a plataforma, é utilizado um labirinto, construído na mesa. Nele existem 3 buracos, sendo 1 de vitória e 2 de derrota. A Figura 28 mostra detalhe da mesa de jogo:



**Figura 28 - Mesa de jogo**

Possui ainda outros sensores indutivos, sendo eles:

CB – sensor carga de bola – avisa o sistema que a bola encontra-se pronta para as operações de inicialização de jogo

- BP – sensor de bola posicionada – informa o sistema que a bola está posicionada e o jogo pode ser iniciado
- SP – sensor perdeu – indica que a bola caiu em algum dos buracos de derrota e o usuário perdeu o jogo
- SG – sensor ganhou – indica que a bola caiu no buraco de vitória e o usuário venceu o jogo

Foram ainda definidos sinais de comunicação com o sistema de mídia, através de chaves ou sinais elétricos equivalentes. As chaves são as seguintes:

- cTE – chave tempo esgotado – indica que o tempo de jogo excedeu o máximo permitido e, portanto, o jogador perdeu.
- cDF – chave dificuldade

A plataforma foi, para este modo de operação, limitada a dois graus de mobilidade (somente as rotações) e estas foram vinculadas à interação do usuário através do joystick.

O usuário, através de um *click* em qualquer um dos botões do joystick, retira o sistema do modo demo (ou permanece no modo jogo, se assim já se encontrar) e o início de jogo ocorre. Para a inicialização do jogo é necessário trazer a esfera (que encontra-se neste momento, invariavelmente, no fundo do canal interno presente na mesa – labirinto – junto ao sensor de carga de bola - CB) para a posição de início de jogo. Para isto, a plataforma inclina-se, fazendo com que a bola se desloque até a parte superior da mesa. Quando é detectada a presença da bola sobre o sensor de bola posicionada (BP), a mesa posiciona-se horizontalmente. Decorre então um tempo de espera de aproximadamente 3 segundos, seguido de uma buzina, que informa ao usuário o início do jogo. Após esta buzina, a plataforma está sob o controle do usuário através do joystick.

Para controlar a mesa a partir do joystick, o PC comunica-se com o joystick e obtém 2 valores digitais, referentes aos encoders internos do joystick e, através de um algoritmo de correção de não-linearidade, são gerados valores de encoders a serem alcançados pelos motores. Além disso, devido à alta taxa de ocorrência de ruídos nos sinais provenientes dos joysticks convencionais, foi feito um filtro de histerese por software, isto é, os valores a serem alcançados pelos motores só são atualizados se os valores recebidos dos encoders do joystick forem tais que a diferença entre eles e os valores dos ciclos de leitura anteriores do joystick forem superiores a um determinado limite pré-estabelecido.

Além disso, diferentemente do modo demo, neste modo, o posicionamento da mesa é cíclico, isto é, os valores de pulsos a serem alcançados pelos motores são atualizados em períodos fixos, e não somente quando o movimento tiver sido finalizado.

Quando o jogo começa, o usuário pode comandar a plataforma através do joystick e para vencer o jogo deve conduzir uma esfera de aço através do labirinto, num tempo limitado, até o buraco de vitória. Se a bola cair no buraco de derrota o usuário perde o controle da mesa, que retorna para a posição horizontal e uma música característica toca. Caso a bola caia no buraco de vitória, acontece o mesmo, tocando uma música diferente, característica de vitória. Há ainda a possibilidade de o tempo de jogo exceder um tempo máximo pré-estabelecido para o jogo (o sistema de mídia da empresa Droid fornece um sinal de chave que indica tempo esgotado – chave TE), sem que a esfera tenha caído em qualquer dos buracos. Neste caso, a música de derrota toca, mas a mesa ainda fica sob comando do joystick, de forma que o próprio usuário ou uma pessoa contratada para o evento possa direcionar a bola para um dos buracos e retornar o sistema para a situação de início de jogo .

Assim como no modo demo, existe um botão (cDF) na parte traseira do gabinete. Este tem a função de regular a dificuldade do jogo, isto é, ele determina o ganho da função de conversão dos encoders do joystick para o número de pulsos passados para os motores. Desta forma, quando o botão está acionado, a dificuldade do jogo é maior, pois a mesa fica mais sensível ao movimento do joystick. Esta função foi implementada conforme pedido pela SEW, prevendo uma possível readequação do número de vencedores e, conseqüentemente, do número de brindes distribuídos.

## 5 - Sistema de posicionamento

É importante salientar que os sistemas mecânico e elétrico montados para funcionamento em exposições e feiras, inclusive o programa que roda nos MoviDrive's, podem ser mantidos os mesmos (rodando no modo demo), sem que, com isto, os objetivos de criar um sistema de posicionamento sejam prejudicados, já que o programa armazenado nos MD's não requer sinais de sensores da mesa (sensores indutivos de presença de bola, essenciais para o funcionamento do modo jogo) nem também valores de chaves e de outros sinais, a não ser os sinais dos encoders e dos sensores de fim de curso, fazendo com que o sistema elétrico não necessite de alterações. Sendo assim, considerando o objetivo principal do projeto de se fazer um sistema de posicionamento modular, utilizável para protótipo de simuladores de voo e para protótipo de centro de usinagem, a única providência necessária para a adequação do sistema é a alteração do software de controle existente no PC. Este pode ser simplificado para que gere números de acordo com um algoritmo diferente apenas, sem que sejam necessárias providências para desabilitar os sensores, fazer ciclos de inicialização (que não o referenciamento) ou qualquer outro procedimento para adequar os MoviDrive's para funcionamento num sistema de posicionamento. Isto desobriga a reprogramação dos MD's, a não ser que se queira que o sistema opere em outro modo de controle pois os MD's podem operar de diversas maneiras diferentes, como por torque controlado, velocidade controlada, etc

## 6 - Resultados obtidos

Os resultados dos trabalhos realizados neste projeto estenderam-se em diversas direções e áreas de pesquisa e aplicação.

Devido às condições de acordo entre a USP e as empresas Phoenix Contact e SEW, uma das abordagens dada a este projeto foi a construção de uma máquina que mostrasse as possibilidades de aplicação dos produtos destas duas empresas em feiras e exposições, através de uma maneira dinâmica e que conseguisse atrair o público. Baseado nestes objetivos foram desenvolvidos conceitos de configuração dos conversores de frequência, adequação mecânica da máquina às condições de aplicação, desenvolvimento de *softwares* de controle, fiação de interconexão de subsistemas, documentações, baterias de testes e conhecimento sobre várias áreas de conhecimento.

Além da máquina de exposição, os objetivos de desenvolvimento de um sistema automático de posicionamento (com aplicação prevista para simuladores de voo e para centros de usinagem) foram obtidos, sendo a máquina desenvolvida perfeitamente adequável a eles, sem que os conceitos básicos tenham que ser repensados (sendo possível inclusive a utilização de conceitos avançados do controle e da fiação que foram desenvolvidos). Isto demonstra que a modularidade esperada para o projeto foi obtida, permitindo grandes vantagens e também uma continuidade e diversificação das áreas de utilização e de estudo que podem ser feitos através da máquina.

## **7 - Conclusões e perspectivas futuras**

O sistema de posicionamento mostrou-se bastante eficaz, respondendo corretamente aos comandos de software em pequeno tempo, e desenvolvendo potência suficiente para muitos dos mais comuns objetivos previstos para a utilização do sistema. O subsistema mecânico, por não ter sido construído com o grau de precisão requerido para uma máquina de posicionamento preciso, não suporta grandes esforços sem sofrer deflexão. Além disto, há direções em que o mecanismo, da maneira como foi projetado, não tem rigidez suficiente, nem mesmo para pequenos esforços.

O funcionamento dos motores, encoders e sistema de comunicação são bastante adequados para aplicação em sistemas de posicionamento preciso, sendo capazes de gerar posicionamento preciso, rigidez mecânica e repetibilidade de posicionamento. Além disto, são capazes de sensoreamento e controle inteligente, através de programas armazenados nos MD`s e possui também a capacidade de comunicação com computadores dotados de placa de comunicação e aquisição de dados compatível com sistema de comunicação adotado neste projeto.

## 8 - Referências Bibliográficas

Tarcísio A. Hess Coelho – Didatic Prototype of a Machine Tool based on a Parallel Kinematic Mechanism – paper code: NTF – 10.

SEW do Brasil – Conversores de Frequência MoviDrive – Catálogo, edição 05 / 99.

SEW Eurodrive – Positioning and Sequence Control IPOS plus, MOVIDRIVE Drive Inverters – Manual, edição 10 / 98.

SEW Eurodrive Operation Instructions, MOVIDRIVE MD\_60A Drive Inverters – Manual, edição 11 / 98.

Cabral, E. L. L. – Análise de Robôs

SEW Eurodrive – Fieldbus Unit Parameter and Data Type Structure Description, MOVIDRIVE Drive Inverters – Manual, edição 06 / 98.

SEW Eurodrive – Operation Instructions, AC Motors AC Brake Motors – Manual, edição 11 / 98.

Boczko, Marcelo; Moraes, Daniel T. B.; Sakura, Roberto – Trabalho Universitário para Escola Politécnica, disciplina PMC-455 - Análise de Sistema de Controle de Nível de Fluido com FieldBus, 1999

## Lista dos Apêndices

- A - Tabela de códigos de falha dos MD's**
- B – Código-fonte do programa do PC em Visual Basic**
- C – Códigos-fonte dos programas dos MD's em IPos Plus®**
- D – Tabelas de Descrição do Projeto de Fiação**
- E – Indicações de Montagem para o *Remote Bus Cable***
- F – Desenhos de Fabricação**
- G – FieldBus**



## Anexo A – Códigos de Falha dos MoviDrive's

O conteúdo deste anexo é de autoria da SEW ®. As tabelas anexas foram obtidas a partir do manual *Positioning and Sequence Control IPOS<sup>Plus</sup>®*, Edição 10/98 disponível pela internet.

| Fault code | Description         | Response             | Possible cause                                                                                                                                                                                                                                                                                                                                                             | Measure                                                                                                                                                                                                                                                                                                                                                                              |
|------------|---------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 00         | No fault            | -                    |                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                      |
| 01         | Overcurrent         | Immediate switch-off | <ul style="list-style-type: none"> <li>- Short-circuit at the output</li> <li>- Motor too large</li> <li>- Faulty output stage</li> </ul>                                                                                                                                                                                                                                  | <ul style="list-style-type: none"> <li>- Remove short-circuit</li> <li>- Connect smaller motor</li> <li>- If fault can still not be reset, contact SEW service</li> </ul>                                                                                                                                                                                                            |
| 03         | Ground fault        | Immediate switch-off | <ul style="list-style-type: none"> <li>- Grounded fault</li> <li>- in the mains connection lead</li> <li>- in the inverter</li> <li>- in the motor</li> </ul>                                                                                                                                                                                                              | <ul style="list-style-type: none"> <li>- Remove ground fault</li> <li>- Contact SEW service</li> </ul>                                                                                                                                                                                                                                                                               |
| 04         | Brake chopper       | Immediate switch-off | <ul style="list-style-type: none"> <li>- Regenerative power too high</li> <li>- Braking resistor circuit interrupted</li> <li>- Short-circuit in the braking resistor circuit</li> <li>- Braking resistance value too high</li> <li>- Faulty brake chopper</li> </ul>                                                                                                      | <ul style="list-style-type: none"> <li>- Increase deceleration ramps</li> <li>- Check connecting leads of the braking resistor</li> <li>- Check technical data of the braking resistor</li> <li>- Exchange MOVIDRIVE®</li> </ul>                                                                                                                                                     |
| 07         | DC link Overvoltage | Immediate switch-off | DC link voltage too high                                                                                                                                                                                                                                                                                                                                                   | <ul style="list-style-type: none"> <li>- Increase deceleration ramps</li> <li>- Check connecting leads of the braking resistor</li> <li>- Check technical data of the braking resistor</li> </ul>                                                                                                                                                                                    |
| 08         | Speed monitoring    | Immediate switch-off | <ul style="list-style-type: none"> <li>- Speed controller resp. current controller (in operating mode VFC without encoder) is at the limit because of mech. overload or phase fault on the supply system or on the motor.</li> <li>- Encoder not connected correctly, (incorrect direction of rotation)</li> <li>- Torque control exceeds <math>n_{max}</math>.</li> </ul> | <ul style="list-style-type: none"> <li>- Reduce load</li> <li>- Increase delay time (P501 resp. P503)</li> <li>- Check encoder connection (Exchange track A/A and B/B in pairs)</li> <li>- Check voltage supply of the encoder</li> <li>- Check current limit</li> <li>- Increase ramps, if required</li> <li>- Check motor and motor leads</li> <li>- Check mains phases</li> </ul> |
| 09         | Commissioning       | Immediate switch-off | Inverter not yet commissioned for selected operating mode                                                                                                                                                                                                                                                                                                                  | Carry out commissioning for selected operating mode                                                                                                                                                                                                                                                                                                                                  |
| 10         | IPOS-ILLOP          | Emergency stop       | <ul style="list-style-type: none"> <li>- Faulty command detected during execution of IPOS program.</li> <li>- Faulty conditions during execution of command.</li> </ul>                                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>- Check contents of the program memory and correct, if required</li> <li>- Download correct program into the program memory</li> <li>- Check program sequence (→ IPOS manual).</li> </ul>                                                                                                                                                     |
| 11         | Overtemperature     | Emergency stop       | Thermal overload of the inverter                                                                                                                                                                                                                                                                                                                                           | Reduce load and / or ensure adequate cooling                                                                                                                                                                                                                                                                                                                                         |
| 12         | Resolver 14 Bit     | Emergency stop       | <b>Only in MDS:</b><br>The 14 Bit resolver evaluation is active and the actual speed is $\geq 3600$ 1/min                                                                                                                                                                                                                                                                  | Set P302 "Maximum speed 1" to max. 3000 1/min.                                                                                                                                                                                                                                                                                                                                       |

| Fault code | Description             | Response             | Possible cause                                                                                                                                                                                                                                                                                                                              | Measure                                                                                                                                                                                                                                                                               |
|------------|-------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13         | Control signal source   | Immediate switch-off | Control signal source not defined or not correctly defined                                                                                                                                                                                                                                                                                  | Select correct control signal source (P101).                                                                                                                                                                                                                                          |
| 14         | Encoder                 | Immediate switch-off | <ul style="list-style-type: none"> <li>Encoder lead or shield not connected correctly</li> <li>Short-circuit/cable breakage in the encoder cable</li> <li>Faulty encoder</li> </ul>                                                                                                                                                         | Check encoder cable and shield for correct connection, short-circuit and cable breakage.                                                                                                                                                                                              |
| 15         | 24 V internal           | Immediate switch-off | Internal 24 V supply voltage missing                                                                                                                                                                                                                                                                                                        | Check mains connection. If the fault cannot be cleared, contact SEW service.                                                                                                                                                                                                          |
| 17-24      | System fault            | Immediate switch-off | Inverter electronics subject to external interference. Possibly due to EMC interference.                                                                                                                                                                                                                                                    | Check connections to ground and shields and improve if required. If the fault cannot be cleared contact SEW service.                                                                                                                                                                  |
| 25         | EEPROM                  | Rapid stop           | Error on accessing the EEPROM                                                                                                                                                                                                                                                                                                               | Call up the factory setting, carry out reset and carry out the commissioning again. If the fault can not be cleared contact SEW service.                                                                                                                                              |
| 26         | External terminal       | Emergency stop       | External fault signal has appeared at a programmable input                                                                                                                                                                                                                                                                                  | Eliminate the cause of the fault. Reprogram the terminal, if necessary.                                                                                                                                                                                                               |
| 27         | Limit switch missing    | Emergency stop       | <ul style="list-style-type: none"> <li>Broken cable or both limit switches missing</li> <li>Limit switches are reversed with respect to the direction motor movement</li> </ul>                                                                                                                                                             | <ul style="list-style-type: none"> <li>Check wiring of the limit switches</li> <li>Exchange limit switch connections or reprogram terminals</li> </ul>                                                                                                                                |
| 28         | Fieldbus timeout        | Rapid stop           | No communication between master and slave within the specified time of the response monitoring.                                                                                                                                                                                                                                             | <ul style="list-style-type: none"> <li>Check communication routine of the master</li> <li>Increase Fieldbus timeout time (P819) 'deactivate monitoring'</li> </ul>                                                                                                                    |
| 29         | Limit switch activated  | Emergency stop       | A limit switch has been approached in <b>IPOS operating mode</b>                                                                                                                                                                                                                                                                            | <ul style="list-style-type: none"> <li>Check travelling range.</li> <li>Correct user program</li> </ul>                                                                                                                                                                               |
| 30         | Emergency stop time-out | Immediate switch-off | <ul style="list-style-type: none"> <li>Drive overload</li> <li>Emergency stop ramp too short</li> </ul>                                                                                                                                                                                                                                     | <ul style="list-style-type: none"> <li>Check project planning</li> <li>Extend emergency stop ramp</li> </ul>                                                                                                                                                                          |
| 31         | Motor TF/TH has tripped | No response          | <ul style="list-style-type: none"> <li>Motor too hot. TF/TH has tripped</li> <li>TF of the motor not connected or not correctly connected.</li> <li>Connection betw. MOVIDRIVE<sup>®</sup> and TF interrupted at the motor.</li> <li>Jumper between X10:1 - X10:2 missing for MDS.</li> <li>Jumper between X15:9 - X15:5 missing</li> </ul> | <ul style="list-style-type: none"> <li>Let motor cool down and reset fault</li> <li>Check connections at betw. MOVIDRIVE<sup>®</sup> and TF.</li> <li>If TF is not connected: Jumper X10:1 with X10:2 For MDS: jumper X15:9 with X15:5.</li> <li>Set P834 to "No response"</li> </ul> |
| 32         | IPOS Index Overflow     | Emergency stop       | Violation of the basic rules for programming, subsequently stack overflow.                                                                                                                                                                                                                                                                  | Check IPOS user program and correct (→ IPOS manual).                                                                                                                                                                                                                                  |
| 33         | Setpoint source         | Immediate switch-off | Setpoint source not / not correctly defined.                                                                                                                                                                                                                                                                                                | Select correct setpoint source (P100).                                                                                                                                                                                                                                                |
| 35         | Operating mode          | Immediate switch-off | Operating mode not / not correctly defined.                                                                                                                                                                                                                                                                                                 | Set correct operating mode with P700 resp. P701                                                                                                                                                                                                                                       |
| 36         | Option not installed    | Immediate switch-off | <ul style="list-style-type: none"> <li>Option pcb not permissible.</li> <li>Setpoint source, control signal source or operating mode not permissible for this option pcb.</li> <li>Encoder type for GPI11A not set correctly</li> </ul>                                                                                                     | <ul style="list-style-type: none"> <li>Insert correct option pcb.</li> <li>Select correct setpoint source (P100)</li> <li>Select correct control signal source (P101)</li> <li>Select correct operating mode (P700 resp. P701).</li> <li>Select correct encoder type.</li> </ul>      |
| 37         | System watchdog         | Immediate switch-off | Fault in the execution of the system software                                                                                                                                                                                                                                                                                               | Contact SEW service.                                                                                                                                                                                                                                                                  |
| 38         | System software         | Immediate switch-off | System error                                                                                                                                                                                                                                                                                                                                | Contact SEW service.                                                                                                                                                                                                                                                                  |
| 39         | Reference travel        | Immediate switch-off | <ul style="list-style-type: none"> <li>Reference cam missing</li> <li>Connection of the limit switches faulty</li> <li>Reference travel type was changed during reference travel.</li> </ul>                                                                                                                                                | Check the reference travel type set and the conditions required                                                                                                                                                                                                                       |

| Fault code | Description          | Response             | Possible cause                                                                                                                                                                                                                                                                                   | Measure                                                                                                                                                                                                                                                                                                                                                           |
|------------|----------------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 40         | Boot synchronization | Immediate switch-off | Only with DIP11A, DPx11A or DRS11A:<br>Fault during boot synchronization between inverter system and option                                                                                                                                                                                      | If fault cannot be cleared exchange option pcb.                                                                                                                                                                                                                                                                                                                   |
| 41         | Watchdog option      | Immediate switch-off | Fault during communication between system software and option software.                                                                                                                                                                                                                          | Contact SEW service.                                                                                                                                                                                                                                                                                                                                              |
| 42         | Lag error            | Immediate switch-off | <ul style="list-style-type: none"> <li>- Polarity of the encoder not set correctly.</li> <li>- Acceleration ramps too short</li> <li>- P-gain of the positioning controller too small</li> <li>- Speed controller not set properly</li> <li>- Value for lag error tolerance too small</li> </ul> | <ul style="list-style-type: none"> <li>- Change encoder polarity</li> <li>- Increase ramps</li> <li>- Increase P-gain</li> <li>- New parameters for speed ctrl.</li> <li>- Increase lag error tolerance</li> <li>- Check wiring of encoder, motor and mains phases</li> <li>- Check if mechanical system is sluggish or whether drive hit an obstacle.</li> </ul> |
| 43         | RS-485 time-out      | Rapid stop           | • Communication between inverter and PC interrupted.                                                                                                                                                                                                                                             | Check connection between inverter and PC. Contact SEW service if necessary.                                                                                                                                                                                                                                                                                       |
| 44         | Unit utilization     | Immediate switch-off | Unit utilization (I <sub>xt</sub> value) higher than 125 %                                                                                                                                                                                                                                       | <ul style="list-style-type: none"> <li>- Reduce power output.</li> <li>- Increase ramps</li> <li>- If above is not possible.: implement larger size inverter</li> </ul>                                                                                                                                                                                           |
| 45         | Initialization       | Immediate switch-off | <ul style="list-style-type: none"> <li>- EEPROM in the power section not or not correctly programmed.</li> <li>- Option board has no connection to Bus</li> </ul>                                                                                                                                | <ul style="list-style-type: none"> <li>- Carry out factory setting. If fault cannot be cleared contact SEW service.</li> <li>- Insert board correctly</li> </ul>                                                                                                                                                                                                  |
| 47         | System bus time-out  | Rapid stop           | • Fault during communication via system bus.                                                                                                                                                                                                                                                     | Check system bus connection.                                                                                                                                                                                                                                                                                                                                      |
| 48         | Hardware DRS         | Immediate switch-off | <ul style="list-style-type: none"> <li>- Encoder signal for master faulty.</li> <li>- Hardware required for synchronous operation is faulty</li> </ul>                                                                                                                                           | <ul style="list-style-type: none"> <li>- Check encoder wiring</li> <li>- Exchange synchronous operation option pcb.</li> </ul>                                                                                                                                                                                                                                    |
| 50         | Pos. HW limit switch | No response          | Only with DPx11A:<br><ul style="list-style-type: none"> <li>- Drive has approached the CW limit switch.</li> <li>- Connection between inverter and CW limit switch interrupted</li> </ul>                                                                                                        | <ul style="list-style-type: none"> <li>- Travel in negative (CCW) direction (e.g. jogging mode).</li> <li>- Determine cause and eliminate.</li> </ul>                                                                                                                                                                                                             |
| 51         | Neg. HW limit switch | No response          | Only with DPx11A:<br><ul style="list-style-type: none"> <li>- Drive has approached the CCW limit switch.</li> <li>- Connection between inverter and CCW limit switch interrupted.</li> </ul>                                                                                                     | <ul style="list-style-type: none"> <li>- Travel in positive (CW) direction (e.g. jogging mode).</li> <li>- Determine cause and eliminate.</li> </ul>                                                                                                                                                                                                              |
| 52         | Pos. SW limit switch | No response          | Only with DPx11A:<br>Target position of the executed travel command lies outside the permissible travel range                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>- Check travelling program and correct if necessary.</li> <li>- Correct position of the CW software limit switch</li> <li>- Switch of monitoring feature by setting both software limit switches to "0".</li> </ul>                                                                                                        |
| 53         | Neg. SW limit switch | No response          | Only with DPx11A:<br>Target position of the executed travel command lies outside the permissible travel range                                                                                                                                                                                    | <ul style="list-style-type: none"> <li>- Check travelling program and correct if necessary.</li> <li>- Correct position of the CCW software limit switch</li> <li>- Switch of monitoring feature by setting both software limit switches to "0".</li> </ul>                                                                                                       |

| Fault code | Description                          | Response    | Possible cause                                                                                                                                                                     | Measure                                                                                                                                 |
|------------|--------------------------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 54         | No home position defined             | No response | <b>Only with DPx11A:</b><br>Reference travel not executed with command "G00" or "SET0".                                                                                            | Perform a reference travel.                                                                                                             |
| 55         | Incorrect machine parameters         | No response | <b>Only with DPx11A:</b><br>Incorrect machine parameters (e.g. wrong value range).                                                                                                 | Check machine parameters and correct.                                                                                                   |
| 56         | Required HW not installed            | No response | <b>Only with DPx11A:</b><br>A not existing hardware is addressed in the in the user program.                                                                                       | Correct user program or insert required hardware into inverter.                                                                         |
| 57         | Selected program does not exist      | No response | <b>Only with DPx11A:</b><br>Attempt to select a non-existing user program.                                                                                                         | – Change program.<br>– Download missing program to the DPx11A.                                                                          |
| 58         | Jump to non-existing line number     | No response | <b>Only with DPx11A:</b><br>Attempt to jump to a non-existing line number in the user program.                                                                                     | Correct user program.                                                                                                                   |
| 59         | Subroutine missing                   | No response | <b>Only with DPx11A:</b><br>Attempt to call a non-existing subroutine in the user program.                                                                                         | – Correct subroutine call in the user program<br>– Download missing subroutine.                                                         |
| 60         | Target position outside travel range | No response | <b>Only with DPx11A:</b><br>Target position of executed command lies outside the permissible travel range.                                                                         | – Correct user program<br>– Adjust travelling range                                                                                     |
| 61         | Prog. speed. > Vmax                  | No response | <b>Only with DPx11A:</b><br>The speed set in the user program is higher than the maximum speed set in the machine parameters.                                                      | – Adjust travel speed in the user program.<br>– Adjust maximum speed in the machine parameters                                          |
| 62         | FLASH-EPROM fault                    | No response | <b>Only with DPx11A:</b><br>Fault during write access on the Flash-EPROM of the DPx11A.                                                                                            | If fault occurs repeatedly contact SEW service.                                                                                         |
| 63         | Division by zero                     | No response | <b>Only with DPx11A:</b><br>Division by zero when calculating SET HxxHyy.                                                                                                          | Correct user program                                                                                                                    |
| 64         | Max. nesting depth reached           | No response | <b>Only with DPx11A:</b><br>– Maximum nesting depth for subroutines reached.<br>– Recursive call of subroutine (Program calls itself)                                              | – Change structure of program<br>– Correct user program.                                                                                |
| 65         | LM628-command error                  | No response | <b>Only with DPx11A:</b><br>Incorrect command to position controller.                                                                                                              | If the fault cannot be cleared or occurs repeatedly contact SEW service.                                                                |
| 66         | Prog. memory full                    | No response | <b>Only with DPx11A:</b><br>Maximum capacity of the program memory exceeded.                                                                                                       | – Delete programs which are no longer required.<br>– If all programs in the program memory are required: optimize contents of programs. |
| 67         | DPx remote line                      | No response | <b>Only with DPx11A:</b><br>Communication in PC controlled mode is interrupted.                                                                                                    | Check connection between PC and inverter.                                                                                               |
| 68         | Target pos not reached               | No response | <b>Only with DPx11A:</b><br>The specified target position was not reached within 5 seconds.<br>– P-gain too small<br>– Position window too small<br>– Travelled drive hit obstacle | – Check setting of the P portion and position window and increase if necessary.<br>– Check for easy movement of the mechanics           |

| Fault code | Description                                            | Response             | Possible cause                                                                                                                                                                                                                                                                                                                  | Measure                                                                                                                                                                                                                                                 |
|------------|--------------------------------------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 69         | Feed enable missing                                    | No response          | Only with DPx11A:<br>Feed enable signal is missing on terminal X11.6.                                                                                                                                                                                                                                                           | Check wiring and signal level at terminal X11.6.                                                                                                                                                                                                        |
| 70         | DPx SSI time-out                                       | No response          | Coded fault, only with DPA11A                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                         |
|            | Code 1: Error SSI interface                            |                      | SSI module faulty                                                                                                                                                                                                                                                                                                               | If the fault cannot be cleared or occurs repeatedly contact SEW service.                                                                                                                                                                                |
|            | Code 2: SSI interface communication error              |                      | SSI module faulty.                                                                                                                                                                                                                                                                                                              | If the fault cannot be cleared or occurs repeatedly contact SEW service.                                                                                                                                                                                |
|            | Code 3: Parity- or power-fail fault of the SSI encoder |                      | <ul style="list-style-type: none"> <li>Encoder lead damaged.</li> <li>Voltage supply interrupted</li> <li>Incorrect machine parameters.</li> </ul>                                                                                                                                                                              | <ul style="list-style-type: none"> <li>Check encoder lead</li> <li>Check voltage supply</li> <li>Check machine parameters and correct if necessary.</li> </ul>                                                                                          |
|            | Code 4: Lag error SSI module                           |                      | Data transfer between encoder and DPA11A interrupted.                                                                                                                                                                                                                                                                           | Check connection lead and corresponding shielding.                                                                                                                                                                                                      |
| 71         | DPx CAN time-out                                       | No response          | Coded fault, only with DPA11A                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                         |
|            | Code 1: Time-out CAN                                   |                      | CAN bus communication interrupted.                                                                                                                                                                                                                                                                                              | Check CAN bus connection.                                                                                                                                                                                                                               |
|            | Code 2: CAN receive buffer full                        |                      | Systematic program fault due to too frequent writing on the CAN bus-interface of an inverter.                                                                                                                                                                                                                                   | Reduce write access to the appropriate inverter in the user program.                                                                                                                                                                                    |
|            | Code 3: CAN controller overflow                        |                      | Malfunction of the CAN controller.                                                                                                                                                                                                                                                                                              | If the fault cannot be cleared or occurs repeatedly contact SEW service.                                                                                                                                                                                |
|            | Code 4: CAN controller error                           |                      | Interference on the CAN bus There might be no bus node available.                                                                                                                                                                                                                                                               | Check wiring as well as user program.                                                                                                                                                                                                                   |
| 72         | Index overflow                                         | No response          | Only with DPx11A:<br>Fault at indexed variable index Offset variable Cxx selected greater than C99.                                                                                                                                                                                                                             | Correct user program.                                                                                                                                                                                                                                   |
| 73         | Illegal operation                                      | No response          | Only with DPx11A:<br>A command, which cannot be executed in the current inverter status, has been issued, e.g. a SAVE command while the drive is travelling.                                                                                                                                                                    | Correct user program                                                                                                                                                                                                                                    |
| 74         | Position outside calculation limit                     | No response          | Only with DPx11A:<br>The calculated setpoint position in increments is greater than $2^{30}$ .                                                                                                                                                                                                                                  | Correct user program.                                                                                                                                                                                                                                   |
| 77         | Invalid IPOS control word                              | No response          | Only in IPOS <sup>plus</sup> operating mode:<br>Attempt to set an invalid automatic mode (via external control)                                                                                                                                                                                                                 | <ul style="list-style-type: none"> <li>Check serial connection to external control.</li> <li>Check writing values of the external control.</li> </ul>                                                                                                   |
| 78         | IPOS SW limit switch                                   | No response          | Only in IPOS operating mode:<br>Target position lies outside the software limit switch travel range.                                                                                                                                                                                                                            | <ul style="list-style-type: none"> <li>Check user program.</li> <li>Check position of the software limit switches.</li> </ul>                                                                                                                           |
| 81         | Starting condition                                     | Immediate switch-off | Only in operating mode "VFC-Holst":<br>The current during premagnetizing period could not be injected in the motor with the required value<br><ul style="list-style-type: none"> <li>Motor rated power selected too small compared to the inverter rated power.</li> <li>Cross-section of the motor cable too small.</li> </ul> | <ul style="list-style-type: none"> <li>Check commissioning data and carry out new commissioning if necessary</li> <li>Check connection between motor and inverter.</li> <li>Check cross-section of the motor cable and increase if necessary</li> </ul> |

| Fault code | Description            | Response             | P Possible cause                                                                                                                                                                                                                                                                                             | Measure                                                                                                                                                                                                |
|------------|------------------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 82         | Output open            | Immediate switch-off | Only in operating mode "VFC-Hoist":<br>- Two or all output phases interrupted.<br>- Motor rated power selected too small compared to the inverter rated power.                                                                                                                                               | - Check connection between motor and inverter.<br>- Check commissioning data and carry out new commissioning if necessary.                                                                             |
| 94         | Motor protection       | Emergency stop       | • Utilization of the motor too high.                                                                                                                                                                                                                                                                         | - Reduce load.<br>- Increase ramps.<br>- Increase duration of rest period.                                                                                                                             |
| 85         | Copy                   | Immediate switch-off | Fault during copying of parameters.                                                                                                                                                                                                                                                                          | Check connection between inverter and PC                                                                                                                                                               |
| 88         | Catch                  | Immediate switch-off | Only in operating mode VFC speed control:<br>Actual speed > 5000 rpm when inverter is enabled                                                                                                                                                                                                                |                                                                                                                                                                                                        |
| 92         | DIP operating range    | Emergency stop       | Only with DIP11A option:<br>Drive has exceeded the permissible operating range of the absolute encoder. Setting of 'Encoder type' / 'Operating range' DIP parameters may be faulty.                                                                                                                          | Check the 'position offset' / 'Zero offset' parameters.                                                                                                                                                |
| 93         | DIP encoder fault      | Emergency stop       | Only with DIP11A option:<br>Encoder reports fault, e.g. power failure.<br>- Encoder-DIP connection cable does not meet requirements (twisted pair cable, shielded).<br>- Switching freq. too high for line length.<br>- Max. permitted speed/acceleration of encoder has been exceeded.<br>- Faulty encoder. | - Check absolute encoder connection<br>- Check connection cable<br>- Set correct switching frequency<br>- Reduce travelling velocity or reduce ramp.<br>- Replace absolute encoder                     |
| 94         | Checksum EEPROM        | Immediate switch-off | Faulty inverter electronics, may be caused by EMC effect or defect.                                                                                                                                                                                                                                          | Send in unit for repair.                                                                                                                                                                               |
| 95         | DIP plausibility error | Emergency stop       | Only with DIP11A option:<br>No plausible position could be determined.<br>- Wrong setting of encoder type.<br>- Wrong setting of iPOS travel parameters.<br>- Wrong setting of numerator/denominator factor.<br>- Zero alignment executed.<br>- Faulty encoder.                                              | - Enter correct encoder type.<br>- Check iPOS travel parameters.<br>- Check travel velocity.<br>- Correct numerator/denominator factor.<br>- Reset after zero alignment<br>- Replace absolute encoder. |

## **Anexo B – Código-fonte do programa do PC em Visual Basic**

O conteúdo deste anexo é a versão do programa de controle com última atualização em 30 de novembro de 2000. Possui os seguintes arquivos:

FormFlu.frm - autoria: Daniel T. B. Moraes e Marcelo Boczko.

Interbus.bas - autoria: Phoenix Contact.

IBSG4HLI.bas - autoria: Phoenix Contact.

Mod\_jogo.bas - autoria: Daniel T. B. Moraes e Marcelo Boczko.

---

### **Arquivo Mod\_jogo.bas**

```
Attribute VB_Name = "Module2"
```

```

Option Explicit
'*****
'Declarações de Mod_Cal
Private erro As Boolean 'Flag de erro
Private aux1 As Boolean 'Auxiliar para autocorreção

'*****
'Declarações de Mod_Flu
Public fim, joystick_habilitado As Boolean
Public modo As Boolean 'modo = true --> Joystick
Public motores_ligados As Boolean
Public pos_JX_anterior, pos_JY_anterior As Integer
Public n1, n2, n3, n4 As Integer
Public aleatorio_ja_desabilitado As Boolean

'*****
Declare Function joyGetDevCaps Lib "MMSYSTEM" (ByVal uJoyID As Integer, lpCaps As JOYCAPS,
ByVal uSize As Integer) As Integer
Declare Function joyGetPos Lib "MMSYSTEM" (ByVal uJoyID As Integer, lpCaps As JOYINFO) As
Integer

Declare Function joySetCapture Lib "MMSYSTEM" (ByVal hWnd As Integer, ByVal uJoyID As Integer,
ByVal uPeriod As Integer, ByVal bChanged As Integer) As Integer
Declare Function joyReleaseCapture Lib "MMSYSTEM" (ByVal uJoyID As Integer) As Integer

Const MAXPNAMELEN = 32

Type JOYCAPS
    wMid As Integer ' manufacturer ID
    wPid As Integer ' product ID
    szPname As String * MAXPNAMELEN ' product name (NULL terminated string)
    wXmin As Long ' minimum x position value
    wXmax As Long ' maximum x position value
    wYmin As Long ' minimum y position value
    wYmax As Long ' maximum y position value
    wZmin As Long ' minimum z position value
    wZmax As Long ' maximum z position value
    wNumButtons As Integer ' number of buttons
    wPeriodMin As Integer ' minimum message period when captured
    wPeriodMax As Integer ' maximum message period when captured
End Type

Type JOYINFO
    wXpos As Integer ' x position
    wYpos As Integer ' y position
    wZpos As Integer ' z position
    wButtons As Integer ' button states
End Type

Dim rt As Long
Dim JoyInformation As JOYINFO
Dim JoyStickCaps As JOYCAPS

'Joystick error codes and return values
Public Const JOYERR_NOERROR = (0) ' no error
Public Const JOYERR_BASE = 160
Public Const JOYERR_UNPLUGGED = (JOYERR_BASE + 7) ' joystick is unplugged
Public Const MMSYSERR_BASE = 0
Public Const MMSYSERR_INVALIDPARAM = (MMSYSERR_BASE + 11) ' invalid parameter passed
Public Const MMSYSERR_NODRIVER = (MMSYSERR_BASE + 6) ' no device driver present
Public Const JOYSTICK1 = 0
Public Const JOYSTICK2 = 1
Public Const JOY_BUTTON1 = &H1
Public Const JOY_BUTTON2 = &H2
Public Const JOY_BUTTON3 = &H4
Public Const JOY_BUTTON4 = &H8

'*****
'Variáveis que guardam a posição do Joystick
Public pos_JX, pos_JY As Integer

```

```

'Flag for ending the game loop
Dim fim_de_loop As Boolean

'Variable to hold the Max Y and Max X values
Dim MaxX As Long
Dim MaxY As Long
'The minimum values
Dim MinX As Long
Dim MinY As Long
Dim kjp As Double 'Constante de conversão Joystick --> pulsos

Public button1_pressed As Boolean
Public button2_pressed As Boolean
Public button3_pressed As Boolean
Public button4_pressed As Boolean

'Variáveis para tratamento de "Tremida do Joystick"
Dim nao_treme1X, nao_treme2X, nao_treme3X As Long
Dim nao_treme1Y, nao_treme2Y, nao_treme3Y As Long
Dim limite_tremedeira As Integer 'Define a instabilidade do joystick
Dim aleatorio_habilitado As Boolean

Public pos1at, pos2at, pos3at, pos4at As Integer
Public dif_pos1, dif_pos2, dif_pos3, dif_pos4 As Integer
Public dif_max As Integer
Public v1, v2, v3, v4 As Integer
*****
' Declarações vindas de Settings
'Valores Default
Const POS_X1f = 1, POS_Y1f = 5, POS_Z1f = -31000, _
    POS_X2f = 2, POS_Y2f = 6, POS_Z2f = -31000, _
    POS_X3f = 3, POS_Y3f = 7, POS_Z3f = -31000, _
    POS_X4f = 4, POS_Y4f = 8, POS_Z4f = -31000
Const POS_MINf = -1000 'Setting parece Ok
Const POS_MAXf = 1000 'Setting parece Ok
Const POS_DIF_34f = 400
Const POS_DIF_1234f = 1000
Const LOAD_AUTOMATICALLYf = False

Public Const VALUE_ALEATORIO = 1500
Public Const MAX_ALE_VEL = 1000

Public POS_DIF_34 As Long ' Máxima diferença entre n3 e n4
Public POS_DIF_1234 As Long ' Máxima diferença entre n1, n2 e n3, n4
Public POS_MIN As Long ' Mínima posição de n1, n2, n3 e n4
Public POS_MAX As Long ' Máxima posição de n1, n2, n3 e n4
    ' Posições de Hastes e Sensores

Public FIM_DE_CURSO_TEMPORARIO As Long '### Deve ser retirada

Public POS_X1 As Long, POS_Y1 As Long, POS_Z1 As Long, _
    POS_X2 As Long, POS_Y2 As Long, POS_Z2 As Long, _
    POS_X3 As Long, POS_Y3 As Long, POS_Z3 As Long, _
    POS_X4 As Long, POS_Y4 As Long, POS_Z4 As Long
Public LOAD_AUTOMATICALLY As Boolean 'Carregar programa na inicialização?
Private k1, k2, k3, k4 As Double 'Conversões de pulsos / mm

Public tempo_parado As Integer
Public tempo_de_jogo As Integer

Public ref1, ref2, ref3, ref4 As Boolean

*****
Dim naux As Integer

Public sensor_ganhou, sensor_perdeu As Boolean
Public sensor_inferior, sensor_superior As Boolean
Public chave_tempo_esgotado As Boolean
Public chave_dificuldade As Boolean
Public tempo_musica As Integer

```

```

Public musicas_de_fim_habilitadas As Boolean
Public Const TEMPO_DESCANSO_PERDEU = 0
Public Const TEMPO_DESCANSO_GANHOU = 24
Public max_joy_vel As Integer

```

```

Public Sub Carrega_Parametros()

```

```

    'OS valores Default devem ainda ser ajustados ###
    POS_X1 = CInt(GetSetting("Plataforma", "Posições", "X1", CStr(POS_X1f)))
    POS_X2 = CInt(GetSetting("Plataforma", "Posições", "X2", CStr(POS_X2f)))
    POS_X3 = CInt(GetSetting("Plataforma", "Posições", "X3", CStr(POS_X3f)))
    POS_X4 = CInt(GetSetting("Plataforma", "Posições", "X4", CStr(POS_X4f)))
    POS_Y1 = CInt(GetSetting("Plataforma", "Posições", "Y1", CStr(POS_Y1f)))
    POS_Y2 = CInt(GetSetting("Plataforma", "Posições", "Y2", CStr(POS_Y2f)))
    POS_Y3 = CInt(GetSetting("Plataforma", "Posições", "Y3", CStr(POS_Y3f)))
    POS_Y4 = CInt(GetSetting("Plataforma", "Posições", "Y4", CStr(POS_Y4f)))
    POS_Z1 = CInt(GetSetting("Plataforma", "Posições", "Z1", CStr(POS_Z1f)))
    POS_Z2 = CInt(GetSetting("Plataforma", "Posições", "Z2", CStr(POS_Z2f)))
    POS_Z3 = CInt(GetSetting("Plataforma", "Posições", "Z3", CStr(POS_Z3f)))
    POS_Z4 = CInt(GetSetting("Plataforma", "Posições", "Z4", CStr(POS_Z4f)))

    POS_DIF_34 = CInt(GetSetting("Plataforma", "Limites", "Diferença_34", CStr(POS_DIF_34f)))
    POS_DIF_1234 = CInt(GetSetting("Plataforma", "Limites", "Diferença_1234", CStr(POS_DIF_1234f)))
    POS_MIN = POS_Z1
    If POS_Z2 > POS_MIN Then
        POS_MIN = POS_Z2
    End If
    If POS_Z3 > POS_MIN Then
        POS_MIN = POS_Z3
    End If
    If POS_Z4 > POS_MIN Then
        POS_MIN = POS_Z4
    End If

    FIM_DE_CURSO_TEMPORARIO = POS_MINf

    POS_MIN = FIM_DE_CURSO_TEMPORARIO
    POS_MAX = POS_MIN + CInt(GetSetting("Plataforma", "Limites", "Curso", CStr(POS_MINf)))
    'aqui é preciso fazer contas, pois não é só posmin ###

    If GetSetting("Plataforma", "Inicialização", "Load Automatically", LOAD_AUTOMATICALLYf) = "yes" Then
        LOAD_AUTOMATICALLY = True
    Else
        LOAD_AUTOMATICALLY = False
    End If
End Sub

```

```

Public Function Corrige_Valor(ByVal Pos As Integer, ByVal min As Integer, ByVal max As Integer) As Integer
    'Esta função impõe limites de movimentos aos fusos
    If Pos < min Then
        Pos = min
    Else
        If Pos > max Then
            Pos = max
        End If
    End If
    Corrige_Valor = Pos
End Function

```

```

Public Sub play_music(numero As Integer)
    Dim valor, valor_antigo As Integer
    Dim i As Integer

```

```

valor_antigo = 0
FormFlu.MMControl1.Command = "Close"
With FormFlu.MMControl1
    .Notify = False
    .Wait = True
    .Shareable = False
    .DeviceType = "Sequencer"
    ' Set the file to be opened before opening the MMControl.
    If numero = 0 Then 'desliga música!!!
        .Command = "close"
    End If
    If numero = 1 Then 'modo Jogo!
        FormFlu.MMControl1.FileName = "C:\MUSICAS\JOGO.MID"
    End If
    If numero = 2 Then 'modo demonstração!
        Do
            valor = Gera_Aleatorio(1, 4)
            Loop While valor = valor_antigo
            valor_antigo = valor
            FormFlu.text8.Text = valor
            Select Case valor
            Case 1
                FormFlu.MMControl1.FileName = "C:\MUSICAS\demo1.MID"
                tempo_musica = 190
            Case 2
                FormFlu.MMControl1.FileName = "C:\MUSICAS\demo2.MID"
                tempo_musica = 102
            Case 3
                FormFlu.MMControl1.FileName = "C:\MUSICAS\demo3.MID"
                tempo_musica = 64
            Case 4
                FormFlu.MMControl1.FileName = "C:\MUSICAS\demo4.MID"
                tempo_musica = 144
            End Select
        End If
        If numero = 3 Then
            'While i < 1000
            '    i = i + 1
            'Wend
            FormFlu.MMControl1.DeviceType = "WaveAudio"
            FormFlu.MMControl1.FileName = "C:\MUSICAS\startup.wav"
            tempo_musica = 0
        End If
        If numero = 4 Then 'perdeu!!!
            While i < 1000
                i = i + 1
            Wend
            FormFlu.MMControl1.FileName = "C:\MUSICAS\perdeu.mid"
            tempo_musica = 10
        End If
        If numero = 5 Then 'ganhou!!!
            While i < 1000
                i = i + 1
            Wend
            FormFlu.MMControl1.FileName = "C:\MUSICAS\ganhou.mid"
            tempo_musica = 10
        End If
        ' Open the MCI WaveAudio device.
        If numero <> 0 Then
            .Command = "Open"
            .Command = "Play"
        End If
    End With
End Sub

```

```

Public Sub Inicializa_Joystick()
'Setting da relação Joystick - Número de pulsos do Encoder

```

```

    kjp = 0.02

'Inicializa variáveis que evitam tremedeira
    nao_treme1X = 10000
    nao_treme2X = 10000
    nao_treme3X = 10000
    nao_treme1Y = 10000
    nao_treme2Y = 10000
    nao_treme3Y = 10000

'Test to see if a Joystick is connected
    rt = joyGetPos(JOYSTICK1, JoyInformation)

    If rt <> JOYERR_NOERROR Then
        If rt = JOYERR_UNPLUGGED Then
            MsgBox "No joystick connected" & vbCrLf & "Finishing..."
        ElseIf rt = MMSYSERR_NODRIVER Then
            MsgBox "No Joystick driver on system" & vbCrLf & "Finishing..."
        Else
            MsgBox "Unknown Error" & vbCrLf & "finishing..."
        End If
        Exit Sub
    End If

'Get the max and min position on the joystick
    joyGetDevCaps JOYSTICK1, JoyStickCaps, Len(JoyStickCaps)

    With JoyStickCaps
        MaxX = 32767
        MinX = -32767
        MaxY = 32000
        MinY = -32767
    End With
    max_joy_vel = 1700
    joystick_habilitado = True
End Sub

Public Sub Joystick_Posiciona(valorx, valory)
    Dim auxX, auxY As Long

    auxX = CInt(kjp * valorx)
    auxY = CInt(kjp * valory)
    Grava_Posicao
    Posiciona auxX, auxX, -auxX - auxY, -auxX + auxY, max_joy_vel, True
'    IBS_HLI_Process_SCL
End Sub

Public Sub Grava_Posicao()
    If Movimento_Finalizado Then
        pos1at = PLAT_1_POS_STA
        pos2at = PLAT_2_POS_STA
        pos3at = PLAT_3_POS_STA
        pos4at = PLAT_4_POS_STA
    End If
End Sub

Function LacoChambre_Joystick(valor, fator As Double) As Long
    If valor >= 0 Then
        LacoChambre_Joystick = valor - 32767
        '32768
    Else
        LacoChambre_Joystick = CInt(valor + 32767) * fator
        '65536
    End If
End Function

Public Sub Le_Joystick()
'Lê joystick para o funcionamento automático
    joyGetPos JOYSTICK1, JoyInformation
    pos_JX = Nao_Treme_X(JoyInformation.wXpos)
    pos_JY = Nao_Treme_Y(JoyInformation.wYpos)

```

```

button1_pressed = ((JoyInformation.wButtons And JOY_BUTTON1) = JOY_BUTTON1)
button2_pressed = ((JoyInformation.wButtons And JOY_BUTTON2) = JOY_BUTTON2)
button3_pressed = ((JoyInformation.wButtons And JOY_BUTTON3) = JOY_BUTTON3)
button4_pressed = ((JoyInformation.wButtons And JOY_BUTTON4) = JOY_BUTTON4)
End Sub

```

'Funções para evitar a tremedeira do Joystick

```

Function Nao_Treme_X(ByVal novo) As Long
    novo = LacoChambre_Joystick(novo, 1.4)
    If ((nao_treme1X - novo < limite_tremedeira) And _
        (novo - nao_treme1X < limite_tremedeira) And _
        (nao_treme2X - novo < limite_tremedeira) And _
        (novo - nao_treme2X < limite_tremedeira) And _
        (nao_treme3X - novo < limite_tremedeira) And _
        (novo - nao_treme3X < limite_tremedeira)) Then
        'Não altera posição (não treme)
        nao_treme2X = nao_treme1X
        nao_treme1X = novo
        novo = nao_treme3X
    Else
        'Altera posição (movimenta)
        nao_treme1X = novo
        nao_treme2X = novo
        nao_treme3X = novo
    End If
    Nao_Treme_X = novo
End Function

```

'Funções para evitar a tremedeira do Joystick

```

Function Nao_Treme_Y(ByVal novo) As Long
    novo = LacoChambre_Joystick(novo, 1)
    If ((nao_treme1Y - novo < limite_tremedeira) And _
        (novo - nao_treme1Y < limite_tremedeira) And _
        (nao_treme2Y - novo < limite_tremedeira) And _
        (novo - nao_treme2Y < limite_tremedeira) And _
        (nao_treme3Y - novo < limite_tremedeira) And _
        (novo - nao_treme3Y < limite_tremedeira)) Then
        'Não altera posição (não treme)
        nao_treme2Y = nao_treme1Y
        nao_treme1Y = novo
        novo = nao_treme3Y
    Else
        'Altera posição (movimenta)
        nao_treme1Y = novo
        nao_treme2Y = novo
        nao_treme3Y = novo
    End If
    Nao_Treme_Y = novo
End Function

```

```

Public Function Botao_Pressionado() As Boolean
    Botao_Pressionado = (button1_pressed Or button2_pressed _
        Or button3_pressed Or button4_pressed)
End Function

```

'As diversas partes são numeradas de acordo

'com numeração do Fluxograma

```

Public Sub Fluxo()
    FormFlu.Show
    'Inicialização
    Inicializacao_Fluxo
    Do
        Passa_Modo_Operacao MoviDrive
        If modo = False Then 'modo automático rodando
            FormFlu.Option1.Value = True
            FormFlu.Option2.Value = False

```

```

        FormFlu.Text1.Text = "Funcionamento Automático"
        Modo_Automatico 'Função que implementa o modo automático
        Le_Joystick
        If Botao_Pressionado() Then
            modo = True
        End If
    Else 'modo Joystick rodando
        FormFlu.Option1.Value = False
        FormFlu.Option2.Value = True
        FormFlu.Text1.Text = "Funcionamento Joystick"
        Modo_Joystick
    End If
Loop While True
'Looping Infinito --> funciona o dia inteiro ou finalizado no botão
End Sub

Public Sub Parte_010()
    tempo_parado = 0
    sensor_perdeu = False
    sensor_ganhou = False
    Do
        Le_Joystick
        DoEvents
        FormFlu.Text1.Text = "Esperando 300 segundos ou botão do joystick"
        FormFlu.Text6.Text = tempo_parado
    Loop While Not ((Botao_Pressionado) Or tempo_parado > 30) '###Mudar para 270 segundos
    If Botao_Pressionado() Then
        modo = True
        'Parte 15
        Passa_Modo_Operacao_MoviDrive
        Movimenta_Inicio_Jogo
        joystick_habilitado = True
    Else
        modo = False 'Modo automático
        'Parte 15
        Passa_Modo_Operacao_MoviDrive
    End If
End Sub

Private Sub Inicializacao_Fluxo()
    limite_tremedeira = 4000
    aleatorio_habilitado = True
    fim = False 'Não terminou o programa
    tempo_parado = 0
    FormFlu.Text6.Text = tempo_parado
    modo = True 'Passa para o modo Joystick

    'Inicializar rede Interbus
    FormFlu.Text1.Text = "Iniciando Rede Interbus"
    IBS_HLI_Init_SCL
    While SCL.BusState <> HLI_IBS_RUN
        IBS_HLI_Init_SCL
    Wend
    FormFlu.TimerInterbus.Enabled = True 'Timer de leitura de Interbus

    'Parte 1, 2 --> Não dependem do programa

    'Parte 3
    FormFlu.Text1.Text = "Aguardando Sinal de PRONTO dos MD's"
    While Not ((Hex(PLAT_1_STATUS And &H100) = 100) And _
        (Hex(PLAT_2_STATUS And &H100) = 100) And _
        (Hex(PLAT_3_STATUS And &H100) = 100) And _
        (Hex(PLAT_4_STATUS And &H100) = 100))
        IBS_HLI_Process_SCL
    Wend

    'Parte 4
    'Envia sinal de PC PRONTO
    FormFlu.Text1.Text = "Enviando Sinal de PC PRONTO p/ MD's"

```

```

PLAT_1_CONTROL = PLAT_1_CONTROL Or &H100
PLAT_2_CONTROL = PLAT_2_CONTROL Or &H100
PLAT_3_CONTROL = PLAT_3_CONTROL Or &H100
PLAT_4_CONTROL = PLAT_4_CONTROL Or &H100
IBS_HLI_Process_SC1

'Parte 5, 6, 7 e 8
FormFlu.Text1.Text = "Referenciando Motores"
Referencia_Motores
motores_ligados = True

'Parte 9, 10, 11, 12, 13
Inicializa_Joystick
FormFlu.Text1.Text = "Aguardando Sinal de Joystick ou 5 minutos"
Parte_910
End Sub

Public Sub Passa_Modo_Operacao_MoviDrive()
FormFlu.Text1.Text = "Enviando Sinal de Modo para MD's"
If modo Then 'Modo Joystick funcionando
'### Palavras de controle parecem erradas
'### Status ou Control??????
    PLAT_1_CONTROL = PLAT_1_CONTROL Or &H400
    PLAT_1_CONTROL = PLAT_1_CONTROL And &HEFFF
    PLAT_2_CONTROL = PLAT_2_CONTROL Or &H400
    PLAT_2_CONTROL = PLAT_2_CONTROL And &HEFFF
    PLAT_3_CONTROL = PLAT_3_CONTROL Or &H400
    PLAT_3_CONTROL = PLAT_3_CONTROL And &HEFFF
    PLAT_4_CONTROL = PLAT_4_CONTROL Or &H400
    PLAT_4_CONTROL = PLAT_4_CONTROL And &HEFFF
Else 'Modo Aleatório
    PLAT_1_CONTROL = PLAT_1_CONTROL And &HEBFF
    PLAT_1_CONTROL = PLAT_1_CONTROL Or &H200
    PLAT_2_CONTROL = PLAT_2_CONTROL And &HEBFF
    PLAT_2_CONTROL = PLAT_2_CONTROL Or &H200
    PLAT_3_CONTROL = PLAT_3_CONTROL And &HEBFF
    PLAT_3_CONTROL = PLAT_3_CONTROL Or &H200
    PLAT_4_CONTROL = PLAT_4_CONTROL And &HEBFF
    PLAT_4_CONTROL = PLAT_4_CONTROL Or &H200
End If
'    IBS_HLI_Process_SC1
End Sub

Public Function Referencia_Motores()
PLAT_1_CONTROL = &H106 'Inicializa Motor 1
PLAT_2_CONTROL = &H106 'Inicializa Motor 2
PLAT_3_CONTROL = &H106 'Inicializa Motor 3
PLAT_4_CONTROL = &H106 'Inicializa Motor 4

'Espera todos referenciarem
ref1 = False
ref2 = False
ref3 = False
ref4 = False
While Not (ref1 And ref2 And ref3 And ref4)
    ref1 = ((PLAT_1_STATUS And &H200) = &H200)
    ref2 = ((PLAT_2_STATUS And &H200) = &H200)
    ref3 = ((PLAT_3_STATUS And &H200) = &H200)
    ref4 = ((PLAT_4_STATUS And &H200) = &H200)
    IBS_HLI_Process_SC1
Wend

'Envia todos para a posição "0" com velocidade constante
PLAT_1_VEL = 1000
PLAT_2_VEL = 1000
PLAT_3_VEL = 1000
PLAT_4_VEL = 1000
PLAT_1_POS = 0
PLAT_2_POS = 0
PLAT_3_POS = 0

```

```

PLAT_4_POS = 0
IBS_HLI_Process_SC1
While (PLAT_1_POS_STA <> PLAT_1_POS Or PLAT_2_POS_STA <> PLAT_2_POS
      Or PLAT_3_POS_STA <> PLAT_3_POS Or PLAT_4_POS_STA <> PLAT_4_POS)
    IBS_HLI_Process_SC1
Wend
End Function

Private Sub Modo_Automatico()
    'Parte 10
    'Inicializa_Motores
    aleatorio_ja_desabilitado = False
    Do
        IBS_HLI_Process_SC1 'Leitura da chave de desabilitação
        aleatorio_habilitado = Not ((PLAT_3_STATUS And &HF00) = &HF00)
        If aleatorio_habilitado Then
            aleatorio_ja_desabilitado = False
            If tempo_musica <= 0 Then
                play_music (2)
            End If
            FormFlu.Text7.Text = tempo_musica
            Le_Joystick

            If Botao_Pressionado() Then
                'Parte 14 não implementada pelo programa
                modo = True
                'Parte 15, 16, 17, 18, 19, 20, 21
                Passa_Modo_Operacao_MoviDrive
                Movimenta_Inicio_Jogo 'Função que coloca bolinha no início do jogo
            Else
                If Movimento_Finalizado() Then
                    n1 = Gera_Aleatorio(POS_MIN, POS_MAX)
                    n2 = Gera_Aleatorio(POS_MIN, POS_MAX)
                    n3 = Gera_Aleatorio(POS_MIN, POS_MAX)
                    n4 = Gera_Aleatorio(POS_MIN, POS_MAX)
                End If
                Grava_Posicao
                erro = Posiciona(n1, n2, n3, n4, MAX_ALE_VEL, True)
            End If
            Else 'Modo aleatório desabilitado
                Le_Joystick
                If Botao_Pressionado() Then
                    'Parte 14 não implementada pelo programa
                    modo = True
                    'Parte 15, 16, 17, 18, 19, 20, 21
                    Passa_Modo_Operacao_MoviDrive
                    Movimenta_Inicio_Jogo 'Função que coloca bolinha no início do jogo
                End If
                If Not aleatorio_ja_desabilitado Then
                    Posiciona 0, 0, 0, 0, MAX_ALE_VEL, False 'Retorna mesa para origem
                    n1 = 0
                    n2 = 0
                    n3 = 0
                    n4 = 0
                    aleatorio_ja_desabilitado = True
                End If
                Posiciona 0, 0, 0, 0, MAX_ALE_VEL, True
                FormFlu.MMControl1.Command = "Close"
                tempo_musica = 1
            End If 'aleatorio habilitado...
            DoEvents 'Dá a preferência a outros eventos
        Loop While modo = False
    End Sub

Function Gera_Aleatorio(limite_inferior As Long, limite_superior As Long) As Long
    Gera_Aleatorio = Int((limite_superior - limite_inferior + 1) * Rnd + limite_inferior)
End Function

Private Sub Modo_Joystick()

```

Do

```
Le_Joystick
'Desligando Borne Início Jogo
If tempo_de_jogo = 3 Then
    PLAT_1_CONTROL = PLAT_1_CONTROL And &HF7FF
    PLAT_2_CONTROL = PLAT_2_CONTROL And &HF7FF
    PLAT_3_CONTROL = PLAT_3_CONTROL And &HF7FF
    PLAT_4_CONTROL = PLAT_4_CONTROL And &HF7FF
    IBS_HLI_Process_SC1
End If

If Movimento() Then
    tempo_parado = 0
    FormFlu.Text6.Text = tempo_parado
    If Not motores_ligados Then
        Inicializa_Motores
        motores_ligados = True
    End If
    If joystick_habilitado Then
        Joystick_Posiciona pos_JX, pos_JY
        chave_dificuldade = ((PLAT_3_STATUS And &H1000) = &H1000)
        If chave_dificuldade Then 'modo difícil acionado
            limite_tremedeira = 4000
            max_joy_vel = 1500
        Else
            limite_tremedeira = 4000
            max_joy_vel = 1800
        End If
    End If
End If

If tempo_parado = 50 Then 'Não mantém ligados os motores
    erro = Finaliza_Motores
    motores_ligados = False
End If

'Parte 23, 24 não dependentes do programa

'Leitura de borne virtual
chave_tempo_esgotado = ((PLAT_1_STATUS And &H2000) = &H2000)
If chave_tempo_esgotado And musicas_de_fim_habilitadas Then
    '###Tocar música de perdeu!!
    play_music (4)
    musicas_de_fim_habilitadas = False
End If

'Parte 25
FormFlu.Text1.Text = "Testando sensor de Armadilha ou tempo esgotado"
sensor_perdeu = (Hex(PLAT_1_STATUS And &H800) = 800)
If sensor_perdeu Then
    joystick_habilitado = False
    '###Tocar música de perdeu!!
    If musicas_de_fim_habilitadas Then
        play_music (4)
    End If
    modo = False
    Passa_Modo_Operacao_MoviDrive
    tempo_parado = 0
    FormFlu.Text6.Text = tempo_parado
    FormFlu.Text1.Text = "Armadilha Acionada"
    Grava_Posicao
    Posiciona 0, 0, 0, 0, MAX_ALE_VEL, True 'Retorna mesa para origem
    Do 'Espera TEMPO_DESCANSO_PERDEU segundos
        DoEvents
        FormFlu.Text6.Text = tempo_parado
    Loop While (tempo_parado < TEMPO_DESCANSO_PERDEU)
    modo = False
    Passa_Modo_Operacao_MoviDrive
    tempo_parado = 0
    FormFlu.Text6.Text = tempo_parado
    FormFlu.Text1.Text = "Reposicionando mesa na origem"
    Grava_Posicao
```

```

        Posiciona 0, 0, 0, 0, MAX_ALE_VEL, True 'Retorna mesa para origem
        Parte_910
    End If
    'Parte 26, 31, 32 Implementado pela mesa de luzes e Movi Drive
    'Parte 33 --> implementado junto com sensor armadilha

    'Parte 27
    sensor_ganhou = (Hex(PLAT_1_STATUS And &H1000) = 1000)
    If sensor_ganhou Then
        FormFlu.Text1.Text = "Vitória acionada"
        'Parte 28
        If musicas_de_fim_habilitadas Then
            play_music (5)
        End If
        Grava_Posicao
        Posiciona 0, 0, 0, 0, MAX_ALE_VEL, True 'Retorna mesa para origem
        joystick_habilitado = False 'Desabilita movimentação da plataforma
        '###Toca música de ganhou!!
        'Parte 29 implementada por Movi Drive

        'Parte 30
        FormFlu.Text1.Text = "Aguardando 30 segundos"
        Do 'Espera TEMPO_DESCANSO_GANHOU segundos
            DoEvents
            FormFlu.Text6.Text = tempo_parado
            Loop While (tempo_parado < TEMPO_DESCANSO_GANHOU)
            modo = False
            Passa_Modo_Operacao_MoviDrive
            tempo_parado = 0
            FormFlu.Text6.Text = tempo_parado
            FormFlu.Text1.Text = "Reposicionando mesa na origem"
            Grava_Posicao
            Posiciona 0, 0, 0, 0, MAX_ALE_VEL, True 'Retorna mesa para origem
            Parte_910
        End If
        DoEvents
        Loop While modo = True
        tempo_parado = 0
        FormFlu.Text6.Text = tempo_parado
    End Sub
    Public Sub Movimenta_Inicio_Jogo()
        FormFlu.Text1.Text = "Modo Jogo"
        play_music (0)
        'Parte 16, 17
        Grava_Posicao
        Posiciona 0, 0, 0, 0, MAX_ALE_VEL, True 'Retorna mesa para origem
        Do 'Espera que bolinha na posição de início de carga
            sensor_inferior = (Hex(PLAT_2_STATUS And &H800) = 800)
            DoEvents
            IBS_HLI_Process_SC1
            Loop While Not sensor_inferior

            FormFlu.Text1.Text = "Aguardando 30 segundos para inicio"
            Do
                DoEvents
                FormFlu.Text6.Text = tempo_parado
            'Do nothing!
            Loop While tempo_parado < 30 'tempo para iniciar o jogo

            'Posicionamento da bolinha no início do jogo - Inclina mesa
            Grava_Posicao
            'Posiciona 400, 400, -300, -300, MAX_ALE_VEL, True
            PLAT_1_VEL = 1000
            PLAT_2_VEL = 1000
            PLAT_3_VEL = 1000
            PLAT_4_VEL = 1000

            PLAT_1_POS = 600
            PLAT_2_POS = 600

```

```

PLAT_3_POS = -400
PLAT_4_POS = -200
'   IBS_HLI_Process_SC1

Do 'Espera que bolinha na posição de início do jogo
    sensor_superior = (Hex(PLAT_2_STATUS And &H1000) = 1000)
    DoEvents
'   IBS_HLI_Process_SC1
Loop While Not sensor_superior

'Posicionamento da bolinha no início do jogo - Desinclina mesa
Grava_Posicao
Posiciona 0, 0, 0, 0, MAX_ALE_VEL, True
'   IBS_HLI_Process_SC1

tempo_parado = 0
FormFlu.Text6.Text = tempo_parado
'Parte 18
FormFlu.Text1.Text = "Aguardando 3 segundos para início"
Do
    DoEvents
    FormFlu.Text6.Text = tempo_parado
    'Do nothing!
Loop While tempo_parado < 3 'tempo para iniciar o jogo
play_music (3) 'Música de Start
While tempo_musica >= 0
    DoEvents
Wend
FormFlu.MMControll.Command = "close"
play_music (1) 'Música de jogo

'Parte 19
'Envia borne virtual inicio_jogo = true
FormFlu.Text1.Text = "Enviando sinal de Início de jogo"
PLAT_1_CONTROL = PLAT_1_CONTROL Or &H800
PLAT_2_CONTROL = PLAT_2_CONTROL Or &H800
PLAT_3_CONTROL = PLAT_3_CONTROL Or &H800
PLAT_4_CONTROL = PLAT_4_CONTROL Or &H800

'Parte 20
modo = True 'Sai do modo automático
'   IBS_HLI_Process_SC1
tempo_parado = 0
FormFlu.Text6.Text = tempo_parado
joystick_habilitado = True
tempo_de_jogo = 0
musicas_de_fim_habilitadas = True
'Parte 21 implementada pelo Movi Drive
'### pode-se colocar musiquinha correspondente aqui
End Sub

Public Function Movimento_Finalizado()
'Dim posicao_lida1, posicao_lida2, posicao_lida3, posicao_lida4 As Integer

'   posicao_lida1 = PLAT_1_POS_STA And &HFF
'   posicao_lida2 = PLAT_2_POS_STA And &HFF
'   posicao_lida3 = PLAT_3_POS_STA And &HFF
'   posicao_lida4 = PLAT_4_POS_STA And &HFF
'   IBS_HLI_Process_SC1
Movimento_Finalizado = (Hex(PLAT_1_STATUS And &H400) = 400) And _
                        (Hex(PLAT_2_STATUS And &H400) = 400) And _
                        (Hex(PLAT_3_STATUS And &H400) = 400) And _
                        (Hex(PLAT_4_STATUS And &H400) = 400)
'### aqui deve haver uma leitura de bornes
End Function

Public Function Movimento() As Boolean
    Movimento = (pos_JX <> pos_JX_anterior Or pos_JY <> pos_JY_anterior)
    pos_JX_anterior = pos_JX
    pos_JY_anterior = pos_JY
End Function

```

```

*** Esta função pretende verificar se o posicionamento é possível,
*** sem prejudicar o equipamento e, neste caso, fazer a chamada
*** ao sistema Interbus, enviando os valores.
*** Em caso de erro, retorna True.
*** Se autocorreção habilitada, não gera resposta de erro e ainda
*** escolhe uma posição possível

Public Function Posiciona2(ByVal pos1 As Long, ByVal pos2 As Long, ByVal pos3 As Long, _
    ByVal pos4 As Long, ByVal velocidade As Long, ByVal autocorrect As Boolean) As Boolean
' Possíveis Problemas
' pos1, pos2, pos3 ou pos4 < posição do sensor
' pos1, pos2, pos3 ou pos4 > posição máxima
' exagerada (positiva ou negativamente) rotação em torno de X (DIF 34)
' exagerada (positiva ou negativamente) rotação em torno de Y (DIF 1234)
erro = False

' Verificação e correção de diferença em pos1, pos2 / pos3, pos4
naux = pos3
If pos3 > pos4 Then
    naux = pos4
End If 'naux é o mínimo de pos3 e pos4
If naux < pos1 - POS_DIF_1234 Then
    If autocorrect Then
        pos1 = naux + POS_DIF_1234
    Else
        erro = True
    End If
End If

naux = pos3
If pos3 < pos4 Then
    naux = pos4
End If 'naux é o máximo de pos3 e pos4
If naux > pos1 + POS_DIF_1234 Then
    If autocorrect Then
        pos1 = naux - POS_DIF_1234
    Else
        erro = True
    End If
End If

' Linha para evitar translação em X
pos2 = pos1

' Verificação e correção de diferença em pos3/pos4
If pos3 < pos4 - POS_DIF_34 Then
    If autocorrect Then
        aux1 = pos4 - POS_DIF_34 - pos3
        pos3 = CInt((pos3 + aux1) / 2)
        pos4 = CInt((pos4 - aux1) / 2)
    Else
        erro = True
    End If
End If
If pos4 < pos3 - POS_DIF_34 Then
    If autocorrect Then
        aux1 = pos3 - POS_DIF_34 - pos4
        pos3 = CInt((pos3 - aux1) / 2)
        pos4 = CInt((pos4 + aux1) / 2)
    Else
        erro = True
    End If
End If

' Verificação e correção de posição muito grande
If pos1 > POS_MAX Then
    If autocorrect Then
        pos1 = POS_MAX
    Else

```

```

        erro = True
    End If
End If
If pos2 > POS_MAX Then
    If autocorrect Then
        pos2 = POS_MAX
    Else
        erro = True
    End If
End If
If pos3 > POS_MAX Then
    If autocorrect Then
        pos3 = POS_MAX
    Else
        erro = True
    End If
End If
If pos4 > POS_MAX Then
    If autocorrect Then
        pos4 = POS_MAX
    Else
        erro = True
    End If
End If

' Verificação e correção de posição muito pequena
If pos1 < POS_MIN Then
    If autocorrect Then
        pos1 = POS_MIN
    Else
        erro = True
    End If
End If
If pos2 < POS_MIN Then
    If autocorrect Then
        pos2 = POS_MIN
    Else
        erro = True
    End If
End If
If pos3 < POS_MIN Then
    If autocorrect Then
        pos3 = POS_MIN
    Else
        erro = True
    End If
End If
If pos4 < POS_MIN Then
    If autocorrect Then
        pos4 = POS_MIN
    Else
        erro = True
    End If
End If
'### Verifique os itens acima...

pos1at = PLAT_1_POS_STA
pos2at = PLAT_2_POS_STA
pos3at = PLAT_3_POS_STA
pos4at = PLAT_4_POS_STA

dif_pos1 = Abs(pos1 - pos1at)
dif_pos2 = Abs(pos2 - pos2at)
dif_pos3 = Abs(pos3 - pos3at)
dif_pos4 = Abs(pos4 - pos4at)

dif_max = 1 'Só para não dividir por zero
If dif_pos1 > dif_max Then
    dif_max = dif_pos1
End If
If dif_pos2 > dif_max Then

```

```

        dif_max = dif_pos2
    End If
    If dif_pos3 > dif_max Then
        dif_max = dif_pos3
    End If
    If dif_pos4 > dif_max Then
        dif_max = dif_pos4
    End If

    v1 = CInt(velocidade * (dif_pos1 / dif_max))
    v2 = CInt(velocidade * (dif_pos2 / dif_max))
    v3 = CInt(velocidade * (dif_pos3 / dif_max))
    v4 = CInt(velocidade * (dif_pos4 / dif_max))

    PLAT_1_VEL = v1
    PLAT_2_VEL = v2
    PLAT_3_VEL = v3
    PLAT_4_VEL = v4

    If v1 < 100 Then
        v1 = 100
    End If
    If v2 < 100 Then
        v2 = 100
    End If
    If v3 < 100 Then
        v3 = 100
    End If
    If v4 < 100 Then
        v4 = 100
    End If

' Posições Verificadas --> envio de informações ao Interbus
    PLAT_1_POS = pos1
    PLAT_2_POS = pos2
    PLAT_3_POS = pos3
    PLAT_4_POS = pos4
' IBS_HMI_Process_S01
    Posiciona2 = erro
End Function

Public Function Posiciona(ByVal pos1 As Long, ByVal pos2 As Long, ByVal pos3 As Long, _
    ByVal pos4 As Long, ByVal velocidade As Long, ByVal autocorrect As Boolean) As Boolean
' Possíveis Problemas
'    pos1, pos2, pos3 ou pos4 < posição do sensor
'    pos1, pos2, pos3 ou pos4 > posição máxima
'    exagerada (positiva ou negativamente) rotação em torno de X (DIF 34)
'    exagerada (positiva ou negativamente) rotação em torno de Y (DIF 1234)
erro = False

' Verificação e correção de diferença em pos1, pos2 / pos3, pos4
pos1 = Corrige_Valor(pos1, -300, 300)
pos2 = Corrige_Valor(pos2, -300, 300)
pos3 = Corrige_Valor(pos3, -300, 300)
pos4 = Corrige_Valor(pos4, -300, 300)

dif_pos1 = Abs(pos1 - pos1at)
dif_pos2 = Abs(pos2 - pos2at)
dif_pos3 = Abs(pos3 - pos3at)
dif_pos4 = Abs(pos4 - pos4at)

dif_max = 1 'Só para não dividir por zero
If dif_pos1 > dif_max Then
    dif_max = dif_pos1
End If
If dif_pos2 > dif_max Then
    dif_max = dif_pos2
End If
If dif_pos3 > dif_max Then

```

```

        dif_max = dif_pos3
    End If
    If dif_pos4 > dif_max Then
        dif_max = dif_pos4
    End If

    v1 = CInt(velocidade * (dif_pos1 / dif_max))
    v2 = CInt(velocidade * (dif_pos2 / dif_max))
    v3 = CInt(velocidade * (dif_pos3 / dif_max))
    v4 = CInt(velocidade * (dif_pos4 / dif_max))

    If v1 < 100 Then
        v1 = 100
    End If
    If v2 < 100 Then
        v2 = 100
    End If
    If v3 < 100 Then
        v3 = 100
    End If
    If v4 < 100 Then
        v4 = 100
    End If

    PLAT_1_VEL = v1
    PLAT_2_VEL = v2
    PLAT_3_VEL = v3
    PLAT_4_VEL = v4

    ' Posições Verificadas --> envio de informações ao Interbus
    PLAT_1_POS = pos1
    PLAT_2_POS = pos2
    PLAT_3_POS = pos3
    PLAT_4_POS = pos4
    '   IBS_HLI_Process_SC1
    '   erro = Posiciona
End Function

Public Function Inicializa_Motores() As Boolean
    IBS_HLI_Init_SC1      'Inicializa Interbus
    FormFlu.TimerInterbus.Enabled = True 'Inicializa Timer de Comunicação Interbus
    PLAT_1_CONTROL = &H6  'Inicializa Motor 1
    PLAT_2_CONTROL = &H6  'Inicializa Motor 2
    PLAT_3_CONTROL = &H6  'Inicializa Motor 3
    PLAT_4_CONTROL = &H6  'Inicializa Motor 4
    Inicializa_Motores = False
    '   IBS_HLI_Process_SC1
End Function

Public Function Finaliza_Motores() As Boolean
    IBS_HLI_Init_SC1
    PLAT_1_CONTROL = &H1  'Finaliza Motor 1
    PLAT_2_CONTROL = &H1  'Finaliza Motor 2
    PLAT_3_CONTROL = &H1  'Finaliza Motor 3
    PLAT_4_CONTROL = &H1  'Finaliza Motor 4
    'For I = 0 To 300000
    'Next 'Temporizador
    '   IBS_HLI_Process_SC1
    Finaliza_Motores = False
End Function

Public Function Finaliza_Interbus() As Boolean
    IBS_HLI_Exit_SC1      'Finaliza Comunicação com Interbus
End Function

```

## Arquivo FormFlu.frm

```

VERSION 5.00
□
Begin VB.Form FormFlu
□
    Caption           = "FormFlu"
□
    ClientHeight      = 6105
    ClientLeft        = 1380
    ClientTop         = 1785
    ClientWidth       = 6690
    LinkTopic         = "Form1"
    PaletteMode       = 1 'UseZOrder
    ScaleHeight       = 6105
    ScaleWidth        = 6690
    Begin VB.TextBox Text8
        Height         = 495
        Left           = 5880
        TabIndex       = 23
        Top            = 5520
        Width          = 615
    End
    Begin VB.TextBox Text7
        Height         = 495
        Left           = 3960
        TabIndex       = 21
        Top            = 5520
        Width          = 1335
    End
    End
    Begin VB.TextBox Text6
        Height         = 495
        Left           = 3960
        TabIndex       = 19
        Top            = 4560
        Width          = 1335
    End
    End
    Begin VB.CommandButton Command2
        Caption        = "Início"
        BeginProperty Font
            Name        = "Arial"
            Size        = 14.25
            Charset     = 0
            Weight      = 400
            Underline   = 0 'False
            Italic      = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Height         = 615
        Left           = 4080
        TabIndex       = 17
        Top            = 2160
        Width          = 1935
    End
    Begin VB.Timer TimerInterbus
        Interval       = 100
        Left           = 960
        Top            = 240
    End
    End
    Begin VB.CommandButton Command1
        Caption        = "Finalizar"
        BeginProperty Font
            Name        = "Arial"
            Size        = 14.25
            Charset     = 0
            Weight      = 400
            Underline   = 0 'False
            Italic      = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Height         = 615
        Left           = 4080
        TabIndex       = 16
        Top            = 2160
        Width          = 1935
    End
    End
    End
    Begin VB.TextBox Text5
        BeginProperty Font
            Name        = "Arial"
            Size        = 12
            Charset     = 0
            Weight      = 400
            Underline   = 0 'False
            Italic      = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Height         = 390
        Left           = 1200
        TabIndex       = 15
        Top            = 5520
        Width          = 1335
    End
    End
    Begin VB.TextBox Text4
        BeginProperty Font
            Name        = "Arial"
            Size        = 12
            Charset     = 0
            Weight      = 400
            Underline   = 0 'False
            Italic      = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Height         = 390
        Left           = 1200
        TabIndex       = 13
        Top            = 4920
        Width          = 1335
    End
    End
    Begin VB.TextBox Text3
        BeginProperty Font
            Name        = "Arial"
            Size        = 12
            Charset     = 0
            Weight      = 400
            Underline   = 0 'False
            Italic      = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Height         = 390
        Left           = 1200
        TabIndex       = 11
        Top            = 4320
        Width          = 1335
    End
    End
    BeginProperty Font
        Name        = "Arial"
        Size        = 12
        Charset     = 0
        Weight      = 400
        Underline   = 0 'False
    EndProperty

```

```

        Italic      = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height      = 390
    Left        = 1200
    TabIndex    = 9
    Top         = 3720
    Width       = 1335
End
Begin VB.OptionButton Option2
    Caption      = "Option1"
    Height       = 255
    Left         = 360
    TabIndex     = 5
    Top          = 2760
    Width        = 255
End
Begin VB.OptionButton Option1
    Caption      = "Option1"
    Height       = 255
    Left         = 360
    TabIndex     = 3
    Top          = 2280
    Width        = 255
End
Begin VB.TextBox Text1
    BeginProperty Font
        Name      = "Times New
Roman"
        Size       = 12
        Charset    = 0
        Weight     = 400
        Underline  = 0 'False
        Italic     = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height       = 495
    Left         = 360
    MaxLength    = 256
    TabIndex     = 1
    TabStop      = 0 'False
    Top          = 840
    Width        = 5655
End
Begin VB.Timer TimerFlu
    Interval     = 1000
    Left         = 360
    Top          = 240
End
Begin VB.PictureBox MMControll
    Height       = 495
    Left         = 2280
    ScaleHeight  = 435
    ScaleWidth   = 3480
    TabIndex     = 20
    Top          = 1560
    Width        = 3540
End
Begin VB.Label Label11
    Caption      = "Tempo Música"
    BeginProperty Font
        Name      = "MS Sans
Serif"
        Size       = 13.5
        Charset    = 0
        Weight     = 700
        Underline  = 0 'False
        Italic     = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height       = 360

```

```

    Left        = 3600
    TabIndex    = 22
    Top         = 5160
    Width       = 2085
End
Begin VB.Label Label10
    Caption      = "Tempo Parado"
    BeginProperty Font
        Name      = "MS Sans
Serif"
        Size       = 13.5
        Charset    = 0
        Weight     = 700
        Underline  = 0 'False
        Italic     = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height       = 360
    Left         = 3600
    TabIndex     = 18
    Top          = 4080
    Width        = 2085
End
Begin VB.Label Label9
    Caption      = "Haste 4"
    BeginProperty Font
        Name      = "Times New
Roman"
        Size       = 12
        Charset    = 0
        Weight     = 400
        Underline  = 0 'False
        Italic     = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height       = 285
    Left         = 240
    TabIndex     = 14
    Top          = 5520
    Width        = 705
End
Begin VB.Label Label8
    Caption      = "Haste 3"
    BeginProperty Font
        Name      = "Times New
Roman"
        Size       = 12
        Charset    = 0
        Weight     = 400
        Underline  = 0 'False
        Italic     = 0 'False
        Strikethrough = 0 'False
    EndProperty
    Height       = 285
    Left         = 240
    TabIndex     = 12
    Top          = 4920
    Width        = 705
End
Begin VB.Label Label7
    Caption      = "Haste 2"
    BeginProperty Font
        Name      = "Times New
Roman"
        Size       = 12
        Charset    = 0
        Weight     = 400
        Underline  = 0 'False
        Italic     = 0 'False
        Strikethrough = 0 'False
    EndProperty

```

```

Height      = 285
Left        = 240
TabIndex    = 10
Top         = 4320
Width       = 705
End
Begin VB.Label Label6
Caption      = "Haste 1"
BeginProperty Font
    Name      = "Times New
Roman"
    Size      = 12
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 285
Left        = 240
TabIndex    = 8
Top         = 3720
Width       = 705
End
Begin VB.Label Label5
Caption      = "Valores"
BeginProperty Font
    Name      = "Arial"
    Size      = 14.25
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 330
Left        = 360
TabIndex    = 7
Top         = 3240
Width       = 1065
End
Begin VB.Label Label4
Caption      = "Modo Jogo"
BeginProperty Font
    Name      = "Times New
Roman"
    Size      = 12
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 285
Left        = 840
TabIndex    = 6
Top         = 2760
Width       = 1065
End
Begin VB.Label Label3
Caption      = "Modo Automático"
BeginProperty Font
    Name      = "Times New
Roman"
    Size      = 12
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty

```

```

Height      = 285
Left        = 840
TabIndex    = 4
Top         = 2280
Width       = 1680
End
Begin VB.Label Label2
Caption      = "Modo Atual"
BeginProperty Font
    Name      = "Arial"
    Size      = 12
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 285
Left        = 480
TabIndex    = 2
Top         = 1800
Width       = 1290
End
Begin VB.Label Label1
Caption      = "Operação Atual"
BeginProperty Font
    Name      = "Arial"
    Size      = 14.25
    Charset   = 0
    Weight    = 700
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Height      = 330
Left        = 2280
TabIndex    = 0
Top         = 480
Width       = 2145
End
End
Attribute VB_Name = "FormFlu"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub Command1_Click()
    Finaliza_Motores
    IBS_HLI_Exit SCI1 'Finaliza
Comunicação Interbus
End
End Sub

Private Sub Command2_Click()
    Fluxo
End Sub

Private Sub Form_Load()
    TimerFlu.Enabled = True
    Carrega_Parametros
    Fluxo
End Sub

Private Sub Timer1_Timer()
    tempo_parado = tempo_parado + 1
    FormFlu.Text6.Text = tempo_parado
End Sub

```

```
Private Sub Option2_Click()  
End Sub  
  
Private Sub TimerFlu_Timer()  
    tempo_parado = tempo_parado + 1  
    tempo_de_jogo = tempo_de_jogo + 1  
    tempo_musica = tempo_musica - 1  
End Sub  
  
Private Sub TimerInterbus_Timer()  
    IBS_HLI_Process_SCL  
End Sub
```

---

## Arquivo Interbus.bas

```
Attribute VB_Name = "Module9"
'*****
'*** IBS G4 HLI      : INTERBUS Controller Initialization Module      **
'*** Application type : Microsoft Visual Basic                      **
'*** Controller-ID    : ISASC1                                       **
'*****
'*** Produced by:                                             **
'*** HLI-Export-Filter Version 1.04 (02/99)                  **
'*** (c) Phoenix Contact 01/1998                             **
'*****

' This array will be set to the bus configuration data
Private IBS_DeviceList(0 To 3) As T_HLI_DEVICE_DATA

' Instance of Controller State Object
Public SC1 As T_HLI_STATE

' private variable to substitute Null reference
Private hliNull As Integer

' Global Process Data Objects
Public PLAT_4_CONTROL      As Integer
Public PLAT_4_VEL         As Integer
Public PLAT_4_POS         As Integer
Public PLAT_4_STATUS      As Integer
Public PLAT_4_VEL_STA     As Integer
Public PLAT_4_POS_STA     As Integer
Public PLAT_3_CONTROL     As Integer
Public PLAT_3_VEL         As Integer
Public PLAT_3_POS         As Integer
Public PLAT_3_STATUS      As Integer
Public PLAT_3_VEL_STA     As Integer
Public PLAT_3_POS_STA     As Integer
Public PLAT_2_CONTROL     As Integer
Public PLAT_2_POS         As Integer
Public PLAT_2_VEL         As Integer
Public PLAT_2_STATUS      As Integer
Public PLAT_2_VEL_STA     As Integer
Public PLAT_2_POS_STA     As Integer
Public PLAT_1_CONTROL     As Integer
Public PLAT_1_VEL         As Integer
Public PLAT_1_POS         As Integer
Public PLAT_1_STATUS      As Integer
Public PLAT_1_VEL_STA     As Integer
Public PLAT_1_POS_STA     As Integer

' function to initialize the device configuration array
Function SC1_SetConfigList(i As Integer, Seg As Integer, Pos As Integer, ID As Integer, Lg As Integer, Lev As Integer, G As Integer, alt As Integer, deac As Integer, st As String) As Integer
Dim z As Integer
IBS_DeviceList(i).Segment = Seg
IBS_DeviceList(i).Position = Pos
IBS_DeviceList(i).ID_Code = ID
IBS_DeviceList(i).LengthCode = Lg
IBS_DeviceList(i).Level = Lev
IBS_DeviceList(i).Group = G
IBS_DeviceList(i).Group = G
IBS_DeviceList(i).Alternative = alt
IBS_DeviceList(i).Deactivated = deac
```

```

For z = 1 To Len(st)
    IBS_DeviceList(i).Station(z - 1) = Asc(Mid(st, z, 1))
Next
IBS_DeviceList(i).Station(z) = 0
End Function

```

```

'#####
'#
'#          Call this function for startup of the INTERBUS system
'#
'#####

```

```

Public Function IBS_HLI_Init_SC1() As Integer
Dim ret As Integer

```

```

' initialize configuration array
Call SC1_SetConfigList(0, 1, 0, 227, 3, 0, 255, 255, &H0, "PLATAFORMA 4")
Call SC1_SetConfigList(1, 2, 0, 227, 3, 0, 255, 255, &H0, "PLATAFORMA 3")
Call SC1_SetConfigList(2, 3, 0, 227, 3, 0, 255, 255, &H0, "PLATAFORMA 2")
Call SC1_SetConfigList(3, 4, 0, 227, 3, 0, 255, 255, &H0, "PLATAFORMA 1")

' call init function
ret = IBS_HLI_Init_CFG(ISASC1, SC1, IBS_STANDARD, 4, IBS_DeviceList(0))
If ret = HLI_OKAY Then
    ' call PDO registration functions
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 1, 0, IBS_PDO_WORD, 0, 0, 16,
    PLAT_4_CONTROL, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 1, 0, IBS_PDO_WORD, 2, 0, 16,
    PLAT_4_VEL, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 1, 0, IBS_PDO_WORD, 4, 0, 16,
    PLAT_4_POS, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 1, 0, IBS_PDO_WORD, 0, 0, 16,
    PLAT_4_STATUS, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 1, 0, IBS_PDO_WORD, 2, 0, 16,
    PLAT_4_VEL_STA, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 1, 0, IBS_PDO_WORD, 4, 0, 16,
    PLAT_4_POS_STA, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 2, 0, IBS_PDO_WORD, 0, 0, 16,
    PLAT_3_CONTROL, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 2, 0, IBS_PDO_WORD, 2, 0, 16,
    PLAT_3_VEL, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 2, 0, IBS_PDO_WORD, 4, 0, 16,
    PLAT_3_POS, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 2, 0, IBS_PDO_WORD, 0, 0, 16,
    PLAT_3_STATUS, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 2, 0, IBS_PDO_WORD, 2, 0, 16,
    PLAT_3_VEL_STA, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 2, 0, IBS_PDO_WORD, 4, 0, 16,
    PLAT_3_POS_STA, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 3, 0, IBS_PDO_WORD, 0, 0, 16,
    PLAT_2_CONTROL, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 3, 0, IBS_PDO_WORD, 4, 0, 16,
    PLAT_2_POS, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 3, 0, IBS_PDO_WORD, 2, 0, 16,
    PLAT_2_VEL, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 3, 0, IBS_PDO_WORD, 0, 0, 16,
    PLAT_2_STATUS, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 3, 0, IBS_PDO_WORD, 2, 0, 16,
    PLAT_2_VEL_STA, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 3, 0, IBS_PDO_WORD, 4, 0, 16,
    PLAT_2_POS_STA, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 4, 0, IBS_PDO_WORD, 0, 0, 16,
    PLAT_1_CONTROL, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 4, 0, IBS_PDO_WORD, 2, 0, 16,
    PLAT_1_VEL, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_OUTPUT, 4, 0, IBS_PDO_WORD, 4, 0, 16,
    PLAT_1_POS, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 4, 0, IBS_PDO_WORD, 0, 0, 16,
    PLAT_1_STATUS, hliNull)

```

```

    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 4, 0, IBS_PDO_WORD, 2, 0, 16,
    PLAT_1_VEL_STA, hliNull)
    Call IBS_HLI_RegisterPDOObject(ISASC1, IBS_PDO_INPUT, 4, 0, IBS_PDO_WORD, 4, 0, 16,
    PLAT_1_POS_STA, hliNull)
    ' Reset all outputs
    Call IBS_HLI_ResetAllOutputs(ISASC1)
    ' Start bus now
    ret = IBS_HLI_StartBus(ISASC1)
End If
IBS_HLI_Init_SC1 = ret

```

End Function

```

'#####
'#
'#           Shut-down of the INTERBUS system -
'#   Call this function before your application terminates !
'#
'#####

```

```

Public Function IBS_HLI_Exit_SC1() As Integer
    IBS_HLI_Exit_SC1 = IBS_HLI_Exit(ISASC1)
End Function

```

```

'#####
'#
'#           Cyclic processing of the INTERBUS system -
'#   This is a PROPOSED FUNCTION for cyclic processing of the
'#   HLI state machine and the INTERBUS process data objects
'#
'#####

```

```

Public Sub IBS_HLI_Process_SC1()
    ' HLI State machine
    Call IBS_HLI_Process(ISASC1)
    ' Look at the Controller-State-object for INTERBUS-State ...

```

```

If SC1.BusState = HLI_IBS_RUN Then

```

```

    ' Update of Interbus-S In-data
    Call IBS_HLI_PD_In(ISASC1)

    ' Here you can place your application-specific code
    ' for process data control */

```

```

    ' Update of Interbus-S Out-data
    Call IBS_HLI_PD_Out(ISASC1)

```

```

End If

```

```

End Sub

```

```

' ----- END OF FILE -----

```

## Arquivo IBSG4HLI.bas

```

Attribute VB_Name = "Module1"
□
'*****
□
'
'
'          PHOENIX CONTACT GmbH & Co., D-32819 Blomberg          **
'          (c) Phoenix Contact 01/1998                          **
'
'*****
'
' Project      : HIGH LEVEL LANGUAGE INTERFACE                  **
'               for PC-based INTERBUS G4 standard controllers (SC series) **
' Sub-Project   : Application Interface for Visual Basic 4 or higher **
'
'*****

'
' File          : IBSG4HLI.BAS
' Component(s)  : -
' Device(s)     : independent
' Terminology  : -
'
' Definition    : S. Gallmann
' Author        : S. Gallmann
'
' Version       : 1.04
' Status        : Released
'
' Description   : Definition/Declaration of Constants, Types, Functions and Class
'               of IBS PC SC HLI for 16-Bit and 32-Bit-Version of Windows
'*****
' Change Notes :
'
' Date          Version      Author      Description
'-----
' 04/98          1.00         SG          Released
' 04/98          1.02         SG          Change in Definition of T_HLI_PCP_DEVICE_INFO;
'                                     Add. Event Code HLI_EVENT_CANNOT_ACTIVATE_CFG
' 09/98          1.03         SG          No changes in this module
' 10/98          1.04         SG          Correction of return value IBS_HLI_GetDiagnosticInfo()

'-----
'
' I.  CONSTANTS
'-----

' Controller board access constants ;
' use these constants for all functions that have the <CID> parameter

' IBS PC ISA SC/I-T Board 1 ... 4
Public Const ISASC1 = &H1
Public Const ISASC2 = &H2
Public Const ISASC3 = &H3
Public Const ISASC4 = &H4

' IBS ETH DSC/I-T Controller 1 ... 4
Public Const ETHDSC1 = &H11
Public Const ETHDSC2 = &H12
Public Const ETHDSC3 = &H13
Public Const ETHDSC4 = &H14

' INTERBUS operating modes **

```

```

' default operating mode of Controller (free running cycles): **
Public Const IBS_NORMAL = 0
Public Const IBS_STANDARD = 0

' application triggers INTERBUS-cycles : **
Public Const IBS_CONTROLLED = 1

' classification of Process-data-object types used for
' PDO registration : **

Public Const IBS_PDO_INPUT = 0          ' input type **
Public Const IBS_PDO_OUTPUT = 1        ' output type **

Public Const IBS_PDO_BOOL = 1           ' bit object **
Public Const IBS_PDO_BITSTRING = 2     ' bit object **
Public Const IBS_PDO_BYTE = 3          ' byte object **
Public Const IBS_PDO_WORD = 4          ' Integer object **
Public Const IBS_PDO_DWORD = 5         ' double Integer object **

' ***** General error codes returned by HLI functions: ***** **

Public Const HLI_OKAY = &H0
Public Const HLI_OK = &H0

Public Const HLI_ACCESS_DENIED = &HFF01
Public Const HLI_INVALID_CID = &HFF02
Public Const HLI_NO_VALID_CFG = &HFF03
Public Const HLI_NO_LOG_WINDOW = &HFF04
Public Const HLI_LOGFILE_OPEN_ERROR = &HFF05

Public Const HLI_MSG_TOO_LONG = &HFF31
Public Const HLI_INVALID_HOST_DATA_ACCESS = &HFF32
Public Const HLI_NO_MESSAGE_AVAILABLE = &HFF33
Public Const HLI_SEND_MESSAGE_TIMEOUT = &HFF34

Public Const HLI_NO_MORE_EVENT = &HFF21

Public Const HLI_PCP_SET_NOT_FOUND = &HFF41
Public Const HLI_PCP_SET_INVALID = &HFF42

' Constants of the <BusState> member of the <T_HLI_STATE> type

Public Const HLI_IBS_READY = &H1        ' dto.          **
Public Const HLI_IBS_ACTIVE = &H2       ' dto.          **
Public Const HLI_IBS_RUN = &H3          ' running cycle(s)**

' Add. states for the application-controlled mode
Public Const HLI_IBS_IDLE = &H4         ' cycle finished **
Public Const HLI_IBS_PREPARING_CYCLE = &H5 ' processing I/O **
Public Const HLI_IBS_CAN_RUN = &H6      ' ready for cycle**

' stop states
Public Const HLI_IBS_STOPPED = &HF001 ' Bus in stop      **
Public Const HLI_IBS_DETECT = &HF002 ' detecting error **
Public Const HLI_IBS_BUS_FAIL = &HF003 ' bus fail->stop   **
Public Const HLI_IBS_CONTROLLER_FAIL = &HF004 ' Controller error **

' constants for PCP device states

Public Const HLI_PCP_DEVICE_READY = 0
Public Const HLI_PCP_DEVICE_NOT_INITIATED = &HF012
Public Const HLI_PCP_DEVICE_BUSY = &HF013
Public Const HLI_PCP_DEVICE_NOT_READY = &HF014

' <State> member constants for PCP service processing

Public Const HLI_PCP_SERVICE_OKAY = &H0

```

```

Public Const HLI_PCP_SERVICE_PENDING = &H1

' States for PCP device programs (<PRG_State> member

Public Const HLI_PCP_PRG_RESET = &H0
Public Const HLI_PCP_PRG_RUNNING = &H1
Public Const HLI_PCP_PRG_STOPPED = &H2

' PCP object types used with PCP-object translation functions

Public Const HLI_POT_BYTE = 0
Public Const HLI_POT_INT16 = 1
Public Const HLI_POT_INT32 = 2
Public Const HLI_POT_FLOAT = 3
Public Const HLI_POT_STRING = 4

' PCP service types received from event parameters

Public Const HLI_PCP_SERVICE_READ = 1
Public Const HLI_PCP_SERVICE_WRITE = 2
Public Const HLI_PCP_SERVICE_IDENTIFY = 3
Public Const HLI_PCP_SERVICE_START = 4
Public Const HLI_PCP_SERVICE_STOP = 5
Public Const HLI_PCP_SERVICE_RESUME = 6
Public Const HLI_PCP_SERVICE_RESET = 7

' ***** HLI Event type definitions *****

' HLI_EG_xxx constants define event groups to use with the
' IBS_HLI_EnableEvents(), IBS_HLI_DisableEvents(),
' IBS_HLI_EnableLogEvents() and IBS_HLI_DisableLogEvents() functions

Public Const HLI_EG_ALL = &HFF

' Event group: internal faults

Public Const HLI_EG_INTERNAL = &H1

Public Const HLI_EVENT_MEMORY_FAULT = &H101
Public Const HLI_EVENT_OS_FAULT = &H102
Public Const HLI_EVENT_NOTE_NO_PF_EV = &H103
Public Const HLI_EVENT_NOTE_NO_PF_RESET_EV = &H104

' Event group: DDI errors

Public Const HLI_EG_DDI = &H2

Public Const HLI_EVENT_DDI_NOT_AVAILABLE = &H201
Public Const HLI_EVENT_DDI_ERROR = &H202
Public Const HLI_EVENT_NO_RIGHTS = &H203
Public Const HLI_EVENT_FW_CNF_TIMEOUT = &H204
Public Const HLI_EVENT_FW_CNF_ERROR = &H205

' Event group: controller faults

Public Const HLI_EG_CTRL = &H4

Public Const HLI_EVENT_CONTROLLER_FAIL = &H401
Public Const HLI_EVENT_SYSTEM_FAIL = &H402
Public Const HLI_EVENT_WATCHDOG_ACTIVE = &H403
Public Const HLI_EVENT_WATCHDOG_SHUT = &H404

' Event group: bus faults

Public Const HLI_EG_BUSEFAULT = &H8

Public Const HLI_EVENT_LOCATED_BUS_ERROR = &H801
Public Const HLI_EVENT_UNLOCATED_BUS_ERROR = &H802
Public Const HLI_EVENT_INVALID_DATA_CYCLE = &H803

```

```

' Event group: device state messages

Public Const HLI_EG_DEVICEMSG = &H10

Public Const HLI_EVENT_DEVICE_ERROR = &H1001
Public Const HLI_EVENT_DEVICE_MAU = &H1002
Public Const HLI_EVENT_DEVICE_RECONFIG = &H1003
Public Const HLI_EVENT_DEVICE_ERROR_RESET = &H1004
Public Const HLI_EVENT_DEVICE_STATE = &H1005

' Event group: bus state changes

Public Const HLI_EG_BUSSTATE = &H20

Public Const HLI_EVENT_BUS_START = &H2001
Public Const HLI_EVENT_BUS_STOP = &H2002
Public Const HLI_EVENT_ALARM_STOP = &H2003
Public Const HLI_EVENT_RB_SEGMENT_OFF = &H2004
Public Const HLI_EVENT_RB_SEGMENT_ON = &H2005
Public Const HLI_EVENT_LB_SEGMENT_OFF = &H2006
Public Const HLI_EVENT_LB_SEGMENT_ON = &H2007
Public Const HLI_EVENT_GROUP_OFF = &H2008
Public Const HLI_EVENT_GROUP_ON = &H2009
Public Const HLI_EVENT_BUS_WARNING = &H200A
Public Const HLI_EVENT_DEVICE_ENABLED = &H200B
Public Const HLI_EVENT_DEVICE_DISABLED = &H200C

' Event group: PCP events

Public Const HLI_EG_PCP = &H40

Public Const HLI_EVENT_PCP_LOCAL_ABORT = &H4001
Public Const HLI_EVENT_PCP_REMOTE_ABORT = &H4002
Public Const HLI_EVENT_PCP_RECONNECTED = &H4003
Public Const HLI_EVENT_PCP_NOT_READY = &H4004

Public Const HLI_EVENT_PCP_SERVICE_FAILED = &H4005
Public Const HLI_EVENT_PCP_SERVICE_REJECTED = &H4006
Public Const HLI_EVENT_PCP_SERVICE_TIMEOUT = &H4007

' Event group: application faults

Public Const HLI_EG_APP = &H80

Public Const HLI_EVENT_NO_VALID_CFG = &H8001
Public Const HLI_EVENT_NULL_POINTER = &H8002
Public Const HLI_EVENT_INVALID_PARA = &H8003

Public Const HLI_EVENT_UNKNOWN_DEVNO = &H8004
Public Const HLI_EVENT_INVALID_IDCODE = &H8005
Public Const HLI_EVENT_NO_PCP_DEVICE = &H8006
Public Const HLI_EVENT_WRONG_NR_OF_DEVICES = &H8007
Public Const HLI_EVENT_UNKNOWN_GROUP = &H8008
Public Const HLI_EVENT_NO_INPUT_DEVICE = &H8009
Public Const HLI_EVENT_NO_OUTPUT_DEVICE = &H800A

Public Const HLI_EVENT_TOO_MANY_PDOS = &H800B
Public Const HLI_EVENT_INVALID_PDO_TYPE = &H800C
Public Const HLI_EVENT_INVALID_PDO_SIZE = &H800D
Public Const HLI_EVENT_INVALID_PDO_POINTER = &H800E
Public Const HLI_EVENT_INVALID_SYNCH_OP = &H800F
Public Const HLI_EVENT_IBS_NOT_RUNNING = &H8010
Public Const HLI_EVENT_IBS_DETECTING = &H8011
Public Const HLI_EVENT_DEVICE_NOT_INITIATED = &H8012
Public Const HLI_EVENT_DEVICE_BUSY = &H8013
Public Const HLI_EVENT_DEVICE_NOT_READY = &H8014

Public Const HLI_EVENT_CFG_INVALID_DEVNO = &H8021
Public Const HLI_EVENT_CFG_INVALID_IDCODE = &H8022
Public Const HLI_EVENT_CFG_INVALID_LEVEL = &H8023
Public Const HLI_EVENT_CFG_INVALID_GROUP = &H8024

```

```

Public Const HLI_EVENT_ACTION_NOT_ALLOWED = &H8031
Public Const HLI_EVENT_CANNOT_ACTIVATE_CFG = &H8032

' **** Constants used by INTERBUS service functions ****

' IBS_HLI_Switchxxx() functions */
Public Const HLI_SEGMENT_OFF = &H0
Public Const HLI_SEGMENT_ON = &H1
Public Const HLI_GROUP_OFF = &H0
Public Const HLI_GROUP_ON = &H1
Public Const HLI_DEVICE_DISABLED = &H0
Public Const HLI_DEVICE_ENABLED = &H1

' Constants for watchdog timeout values used by the
' IBS_HLI_ActivateWatchdog() function
Public Const HLI_WDT_8 = &H0           ' Timeout = 8.2 ms **
Public Const HLI_WDT_16 = &H4          ' Timeout = 16.4 ms **
Public Const HLI_WDT_32 = &H8          ' Timeout = 32.8 ms **
Public Const HLI_WDT_65 = &HC          ' Timeout = 65.5 ms **
Public Const HLI_WDT_131 = &H10        ' Timeout = 131.1 ms **
Public Const HLI_WDT_262 = &H14        ' Timeout = 262.1 ms **
Public Const HLI_WDT_524 = &H18        ' Timeout = 524.3 ms **
Public Const HLI_WDT_1048 = &H1C       ' Timeout = 1048.6 ms **

' ***** Event text & window constants ***** **

Public Const HLI_ENGLISH = &H0
Public Const HLI_DEUTSCH = &H1
Public Const HLI_EW_FIXED = &H0
Public Const HLI_EW_FILTERCHANGE = &H1

' HOST-COP-communication

Public Const DATA_AREA_SIZE = 5120

' -----**
' 11. TYPE DEFINITIONS **
' -----**

' struct to hold HLI state information **
'*****

' record to hold HLI state information

Type T_HLI_STATE
    BusState As Integer
    DiagState As Integer
    EventIndication As Integer
End Type

' record to hold HLI date/time information

Type T_HLI_TIME
    d As Integer
    mo As Integer
    y As Integer
    h As Integer
    m As Integer
    s As Integer
End Type

' record to hold HLI event information

Type T_HLI_EVENT
    Timestamp As T_HLI_TIME

```

```

        CID           As Integer
        Code          As Integer
        Param(0 To 4) As Integer
    End Type

```

' record to controller information

```

Type T_HLI_SC_INFO

```

```

    reserved1 As Byte
    Ident     As String * 32
    reserved2 As Byte
    FW_Version As String * 8
    reserved3 As Byte
    HW_Version As String * 5
End Type

```

' record to hold Bus information

```

Type T_HLI_BUS_INFO

```

```

    Devices      As Integer
    IO_Points     As Integer
    PCP_Devices   As Integer
    PD_Length     As Integer
    Cychetime     As Long
End Type

```

' record to hold bus device data

```

Type T_HLI_DEVICE_DATA

```

```

    Segment      As Integer
    Position      As Integer
    ID_Code       As Integer
    LengthCode    As Integer
    Level         As Integer
    Group         As Integer
    Alternative    As Integer
    Deactivated   As Integer
    Station(0 To 31) As Byte
End Type

```

' record to hold bus device information

```

Type T_HLI_DEVICE_INFO

```

```

    Segment      As Integer
    Position      As Integer
    ID_Code       As Integer
    LengthCode    As Integer
    Level         As Integer
    Group         As Integer
    Alternative    As Integer
    InBitLength   As Integer
    OutBitLength  As Integer
    PCP_Ck        As Integer
    Active        As Integer
End Type

```

' record to hold PCP device information

```

Type T_HLI_PCP_DEVICE_INFO

```

```

    Segment      As Integer
    Position      As Integer
    State         As Integer
    ID_Code       As Integer
    LengthCode    As Integer
    reserved1     As Byte
    Manufacturer   As String * 255

```

```

        reserved2      As Byte
        Name           As String * 255
        reserved3      As Byte
        Revision       As String * 255
    End Type

' record to hold pcg data object

Type T_HLI_PCP_RW

    Segment           As Integer
    Position          As Integer
    Index             As Integer
    Subindex          As Integer
    length            As Integer
    State             As Integer
    Data(0 To 255) As Byte
End Type

' record to hold pcg program object

Type T_HLI_PCP_PRG

    Segment           As Integer
    Position          As Integer
    Index             As Integer
    PRG_State         As Integer
    State             As Integer
End Type

' record to diagnostic information

Type T_HLI_SC_DIAG

    Cycle_Count       As Long
    Cycle_Error_Count As Long
    ID_Cycle_Count     As Long
    ID_Cycle_Error_Count As Long
    Data_Cycle_Count   As Long
    Data_Cycle_Error_Count As Long
End Type

' record to hold message block for COP communication

Type T_HLI_MSG

    Code              As Integer
    nParms            As Integer
    Parms(0 To 511) As Integer
End Type

' -----**
'           H L I  FUNCTIONS                      **
' -----**

' ##### Windows 16-Bit-Version #####

#If Win16 Then

Public Declare Function IBS_HLI_GetVersion Lib "G4HLIW16.DLL" (ByVal destination As String) As Integer

' HLI-Init-functions

```

```

Public Declare Function IBS_HLI_Init Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByRef SC As
T_HLI_STATE, ByVal ibs_mode As Integer) As Integer
Public Declare Function IBS_HLI_Init_CFG Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByRef SC As
T_HLI_STATE, ByVal ibs_mode As Integer, ByVal nDevices As Integer, ByRef DeviceList As
T_HLI_DEVICE_DATA) As Integer
Public Declare Function IBS_HLI_Exit Lib "G4HLIW16.DLL" (ByVal CID As Integer) As Integer

' HLI-Event management

Public Declare Function IBS_HLI_EnableEvents Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
eventgroup As Integer) As Integer
Public Declare Function IBS_HLI_DisableEvents Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
eventgroup As Integer) As Integer
Public Declare Function IBS_HLI_GetEventInfo Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByRef
Ev As T_HLI_EVENT) As Integer
Public Declare Function IBS_HLI_GetEventText Lib "G4HLIW16.DLL" (ByRef Ev As T_HLI_EVENT,
ByVal Language As Integer, ByVal destination As String) As Integer
Public Declare Function IBS_HLI_EnableLogEvents Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal eventgroup As Integer) As Integer
Public Declare Function IBS_HLI_DisableLogEvents Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal eventgroup As Integer) As Integer
Public Declare Function IBS_HLI_CreateLogWindow Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal Language As Integer, ByVal FilterChange As Integer) As Integer
Public Declare Function IBS_HLI_PopupLogWindow Lib "G4HLIW16.DLL" (ByVal CID As Integer) As
Integer
Public Declare Function IBS_HLI_CloseLogWindow Lib "G4HLIW16.DLL" (ByVal CID As Integer) As
Integer
Public Declare Function IBS_HLI_LogUserEvent Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
UserText As String, ByVal WithTimeStamp As Boolean) As Integer
Public Declare Function IBS_HLI_LogAutoSave Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
Filename As String, ByVal Append As Boolean) As Integer

' HLI-INTERBUS-Information functions

Public Declare Function IBS_HLI_GetControllerInfo Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByRef CI As T_HLI_SC_INFO) As Integer
Public Declare Function IBS_HLI_GetBusInfo Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByRef BI
As T_HLI_BUS_INFO) As Integer
Public Declare Function IBS_HLI_GetDeviceInfo Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
Index As Integer, ByRef DI As T_HLI_DEVICE_INFO) As Integer
Public Declare Function IBS_HLI_GetDiagnosticInfo Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByRef DI As T_HLI_SC_DIAG) As Integer

' HLI-INTERBUS-Management-functions

' HLI-State-Machine-function
Public Declare Function IBS_HLI_Process Lib "G4HLIW16.DLL" (ByVal CID As Integer) As Integer

' HLI-process data management
Public Declare Function IBS_HLI_PD_In Lib "G4HLIW16.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_PD_Out Lib "G4HLIW16.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_PD_Input Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByRef
lpAppVar As Any) As Integer
Public Declare Function IBS_HLI_PD_Output Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByRef
lpAppVar As Any) As Integer
Public Declare Function IBS_HLI_PD_DeviceIn Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
segment_nr As Integer, ByVal segment_pos As Integer) As Integer
Public Declare Function IBS_HLI_PD_DeviceOut Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
segment_nr As Integer, ByVal segment_pos As Integer) As Integer
Public Declare Function IBS_HLI_ResetAllOutputs Lib "G4HLIW16.DLL" (ByVal CID As Integer) As
Integer

' for application controlled mode only
Public Declare Function IBS_HLI_GetCyclePhase Lib "G4HLIW16.DLL" () As Integer
Public Declare Function IBS_HLI_RunCycle Lib "G4HLIW16.DLL" () As Integer
Public Declare Function IBS_HLI_RunCompleteCycle Lib "G4HLIW16.DLL" () As Integer

' INTERBUS-PD-Object registration function

' master function

```

```

Public Declare Function IBS_HLI_RegisterPDOObject Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
PDOType As Integer, ByVal ByteOffset As Integer, ByVal BitPosition As Integer, ByVal length As
Integer, ByRef lpAppVar As Any, ByRef lpChanged As Any) As Integer
' Bool objects
Public Declare Function IBS_HLI_RegisterPDO_BOOL Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
ByteOffset As Integer, ByVal BitPosition As Integer, ByRef lpAppVar As Integer, ByRef
lpChanged As Any) As Integer

' Bitstring objects
Public Declare Function IBS_HLI_RegisterPDO_BITSTRING Lib "G4HLIW16.DLL" (ByVal CID As
Integer, ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer,
ByVal ByteOffset As Integer, ByVal BitPosition As Integer, ByVal BitLength As Integer, ByRef
lpAppVar As Integer, ByRef lpChanged As Any) As Integer

' Integer objects
Public Declare Function IBS_HLI_RegisterPDO_BYTE Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
ByteOffset As Integer, ByRef lpAppVar As Byte, ByRef lpChanged As Any) As Integer

' Word objects
Public Declare Function IBS_HLI_RegisterPDO_WORD Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
ByteOffset As Integer, ByRef lpAppVar As Integer, ByRef lpChanged As Any) As Integer

' DWord objects
Public Declare Function IBS_HLI_RegisterPDO_DWORD Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
ByteOffset As Integer, ByRef lpAppVar As Long, ByRef lpChanged As Any) As Integer

' unregistering objects
Public Declare Function IBS_HLI_UnregisterPDOObject Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByRef lpAppVar As Any) As Integer

' INTERBUS-PCP functions

Public Declare Function IBS_HLI_PCP_SetConnectTimeout Lib "G4HLIW16.DLL" (ByVal CID As
Integer, ByVal timeout As Integer) As Integer
Public Declare Function IBS_HLI_PCP_DeviceState Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal Segment As Integer, ByVal Position As Integer) As Integer
Public Declare Function IBS_HLI_PCP_ReconnectDevice Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal Segment As Integer, ByVal Position As Integer) As Integer
Public Declare Function IBS_HLI_PCP_ReadRequest Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpRRQ As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_ReadService Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpRRQ As T_HLI_PCP_RW, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_WriteRequest Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpWRQ As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_WriteService Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpWRQ As T_HLI_PCP_RW, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_IdentifyRequest Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpDI As T_HLI_PCP_DEVICE_INFO) As Integer
Public Declare Function IBS_HLI_PCP_IdentifyService Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpDI As T_HLI_PCP_DEVICE_INFO, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_RegisterClientObject Lib "G4HLIW16.DLL" (ByVal CID As
Integer, lpWRQ As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_StartRequest Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG) As Integer
Public Declare Function IBS_HLI_PCP_StartService Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_StopRequest Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG) As Integer
Public Declare Function IBS_HLI_PCP_StopService Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_ResumeRequest Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG) As Integer
Public Declare Function IBS_HLI_PCP_ResumeService Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_ResetRequest Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG) As Integer

```

```
Public Declare Function IBS_HLI_PCP_ResetService Lib "G4HLIW16.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG, timeout_ms As Integer) As Integer
```

```
' INTERBUS-PCP-object translation functions
```

```
Public Declare Function IBS_HLI_PCP_LoadObjectByName Lib "G4HLIW16.DLL" (ByVal Filename As
String, ByVal VarName As String, ByRef lpPCPObj As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_LoadObjectByIndex Lib "G4HLIW16.DLL" (ByVal Filoname As
String, ByVal Index As Integer, ByRef lpPCPObj As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_LoadObjectByEntry Lib "G4HLIW16.DLL" (ByVal Filename As
String, ByVal Entry As Integer, ByRef lpPCPObj As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_InitObject Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Segment As Integer, ByVal Position As Integer, ByVal Index As Integer,
ByVal Subindex As Integer, ByVal POT As Integer, ByVal nElements As Integer, ByRef lpData As
Any) As Integer
Public Declare Function IBS_HLI_PCPObject_SetInt16 Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Offset As Integer, ByVal VInt16 As Integer) As Integer
Public Declare Function IBS_HLI_PCPObject_SetInt32 Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Offset As Integer, ByVal VInt32 As Long) As Integer
Public Declare Function IBS_HLI_PCPObject_SetFloat Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Offset As Integer, ByVal VFloat As Single) As Integer
Public Declare Function IBS_HLI_PCPObject_SetString Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Offset As Integer, ByVal lpS As String) As Integer
```

```
Public Declare Function IBS_HLI_PCPObject_GetInt16 Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Offset As Integer, ByRef lpInt16 As Integer) As Integer
Public Declare Function IBS_HLI_PCPObject_GetInt32 Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Offset As Integer, ByRef lpInt32 As Long) As Integer
Public Declare Function IBS_HLI_PCPObject_GetFloat Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Offset As Integer, ByRef lpFloat As Single) As Integer
Public Declare Function IBS_HLI_PCPObject_GetString Lib "G4HLIW16.DLL" (ByRef lpPCPObj As
T_HLI_PCP_RW, ByVal Offset As Integer, ByVal lpS As String) As Integer
```

```
' Controller-Services
```

```
Public Declare Function IBS_HLI_ActivateWatchdog Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal WDT As Integer) As Integer
```

```
' INTERBUS-Services
```

```
Public Declare Function IBS_HLI_StartBus Lib "G4HLIW16.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_StopBus Lib "G4HLIW16.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_AlarmStop Lib "G4HLIW16.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_SwitchRemoteBus Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal segment_nr As Integer, ByVal switch_code As Integer) As Integer
Public Declare Function IBS_HLI_SwitchLocalBus Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
segment_nr As Integer, ByVal switch_code As Integer) As Integer
Public Declare Function IBS_HLI_SwitchGroup Lib "G4HLIW16.DLL" (ByVal CID As Integer, ByVal
group_nr As Integer, ByVal Alternative As Integer, ByVal switch_code As Integer) As Integer
Public Declare Function IBS_HLI_ChangeDeviceState Lib "G4HLIW16.DLL" (ByVal CID As Integer,
ByVal Segment As Integer, ByVal Position As Integer, ByVal new_state As Integer) As Integer
```

```
#End If
```

```
' ##### Windows 32-Bit-Version #####
```

```
#If Win32 Then
```

```
Public Declare Function IBS_HLI_GetVersion Lib "G4HLIW32.DLL" (ByVal destination As String) As
Integer
```

```
' HLI-Init-functions
```

```
Public Declare Function IBS_HLI_Init Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByRef SC As
T_HLI_STATE, ByVal ibs_mode As Integer) As Integer
Public Declare Function IBS_HLI_Init_CFG Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByRef SC As
T_HLI_STATE, ByVal ibs_mode As Integer, ByVal nDevices As Integer, ByRef DeviceList As
T_HLI_DEVICE_DATA) As Integer
```

```

Public Declare Function IBS_HLI_Exit Lib "G4HLIW32.DLL" (ByVal CID As Integer) As Integer

' HLI-Event management

Public Declare Function IBS_HLI_EnableEvents Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal
eventgroup As Integer) As Integer
Public Declare Function IBS_HLI_DisableEvents Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal
eventgroup As Integer) As Integer
Public Declare Function IBS_HLI_GetEventInfo Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByRef
Ev As T_HLI_EVENT) As Integer
Public Declare Function IBS_HLI_GetEventText Lib "G4HLIW32.DLL" (ByRef Ev As T_HLI_EVENT,
ByVal Language As Integer, ByVal destination As String) As Integer
Public Declare Function IBS_HLI_EnableLogEvents Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal eventgroup As Integer) As Integer
Public Declare Function IBS_HLI_DisableLogEvents Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal eventgroup As Integer) As Integer
Public Declare Function IBS_HLI_CreateLogWindow Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal Language As Integer, ByVal FilterChange As Integer) As Integer
Public Declare Function IBS_HLI_PopupLogWindow Lib "G4HLIW32.DLL" (ByVal CID As Integer) As
Integer
Public Declare Function IBS_HLI_CloseLogWindow Lib "G4HLIW32.DLL" (ByVal CID As Integer) As
Integer
Public Declare Function IBS_HLI_LogUserEvent Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal
UserText As String, ByVal WithTimeStamp As Boolean) As Integer
Public Declare Function IBS_HLI_LogAutoSave Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal
Filename As String, ByVal Append As Boolean) As Integer

' HLI-INTERBUS-information functions

Public Declare Function IBS_HLI_GetControllerInfo Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByRef CI As T_HLI_SC_INFO) As Integer
Public Declare Function IBS_HLI_GetBusInfo Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByRef BI
As T_HLI_BUS_INFO) As Integer
Public Declare Function IBS_HLI_GetDeviceInfo Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal
Index As Integer, ByRef DI As T_HLI_DEVICE_INFO) As Integer
Public Declare Function IBS_HLI_GetDiagnosticInfo Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByRef DI As T_HLI_SC_DIAG) As Integer

' HLI-INTERBUS-Management-functions
' HLI-State-Machine-function
Public Declare Function IBS_HLI_Process Lib "G4HLIW32.DLL" (ByVal CID As Integer) As Integer

' HLI-process data management
Public Declare Function IBS_HLI_PD_In Lib "G4HLIW32.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_PD_Out Lib "G4HLIW32.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_PD_Input Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByRef
lpAppVar As Any) As Integer
Public Declare Function IBS_HLI_PD_Output Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByRef
lpAppVar As Any) As Integer
Public Declare Function IBS_HLI_PD_DeviceIn Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal
segment_nr As Integer, ByVal segment_pos As Integer) As Integer
Public Declare Function IBS_HLI_PD_DeviceOut Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal
segment_nr As Integer, ByVal segment_pos As Integer) As Integer
Public Declare Function IBS_HLI_ResetAllOutputs Lib "G4HLIW32.DLL" (ByVal CID As Integer) As
Integer

' for application controlled mode only
Public Declare Function IBS_HLI_GetCyclePhase Lib "G4HLIW32.DLL" () As Integer
Public Declare Function IBS_HLI_RunCycle Lib "G4HLIW32.DLL" () As Integer
Public Declare Function IBS_HLI_RunCompleteCycle Lib "G4HLIW32.DLL" () As Integer

' INTERBUS-PD-Object registration function

' master function
Public Declare Function IBS_HLI_RegisterPDObject Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
PDOType As Integer, ByVal ByteOffset As Integer, ByVal BitPosition As Integer, ByVal length As
Integer, ByRef lpAppVar As Any, ByRef lpChanged As Any) As Integer

' Bool objects

```

```

Public Declare Function IBS_HLI_RegisterPDO_BOOL Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
ByteOffset As Integer, ByVal BitPosition As Integer, ByRef lpAppVar As Integer, ByRef
lpChanged As Any) As Integer

' Bitstring objects
Public Declare Function IBS_HLI_RegisterPDO_BITSTRING Lib "G4HLIW32.DLL" (ByVal CID As
Integer, ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer,
ByVal ByteOffset As Integer, ByVal BitPosition As Integer, ByVal BitLength As Integer, ByRef
lpAppVar As Integer, ByRef lpChanged As Any) As Integer

' Integer objects
Public Declare Function IBS_HLI_RegisterPDO_BYTE Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
ByteOffset As Integer, ByRef lpAppVar As Byte, ByRef lpChanged As Any) As Integer

' Word objects
Public Declare Function IBS_HLI_RegisterPDO_WORD Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
ByteOffset As Integer, ByRef lpAppVar As Integer, ByRef lpChanged As Any) As Integer

' DWord objects
Public Declare Function IBS_HLI_RegisterPDO_DWORD Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal PD_Type As Integer, ByVal segment_nr As Integer, ByVal segment_pos As Integer, ByVal
ByteOffset As Integer, ByRef lpAppVar As Long, ByRef lpChanged As Any) As Integer

' unregistering objects
Public Declare Function IBS_HLI_UnregisterPDOObject Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByRef lpAppVar As Any) As Integer

' INTERBUS-PCP functions
Public Declare Function IBS_HLI_PCP_SetConnectTimeout Lib "G4HLIW32.DLL" (ByVal CID As
Integer, ByVal timeout As Integer) As Integer
Public Declare Function IBS_HLI_PCP_DeviceState Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal Segment As Integer, ByVal Position As Integer) As Integer
Public Declare Function IBS_HLI_PCP_ReconnectDevice Lib "G4HLIW32.DLL" (ByVal CID As Integer,
ByVal Segment As Integer, ByVal Position As Integer) As Integer
Public Declare Function IBS_HLI_PCP_ReadRequest Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpRRQ As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_ReadService Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpRRQ As T_HLI_PCP_RW, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_WriteRequest Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpWRQ As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_WriteService Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpWRQ As T_HLI_PCP_RW, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_IdentifyRequest Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpDI As T_HLI_PCP_DEVICE_INFO) As Integer
Public Declare Function IBS_HLI_PCP_IdentifyService Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpDI As T_HLI_PCP_DEVICE_INFO, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_RegisterClientObject Lib "G4HLIW32.DLL" (ByVal CID As
Integer, lpWRQ As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_StartRequest Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG) As Integer
Public Declare Function IBS_HLI_PCP_StartService Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_StopRequest Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG) As Integer
Public Declare Function IBS_HLI_PCP_StopService Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_ResumeRequest Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG) As Integer
Public Declare Function IBS_HLI_PCP_ResumeService Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG, timeout_ms As Integer) As Integer
Public Declare Function IBS_HLI_PCP_ResetRequest Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG) As Integer
Public Declare Function IBS_HLI_PCP_ResetService Lib "G4HLIW32.DLL" (ByVal CID As Integer,
lpPRG As T_HLI_PCP_PRG, timeout_ms As Integer) As Integer

' INTERBUS-PCP-object translation functions

```

```

Public Declare Function IBS_HLI_PCP_LoadObjectByName Lib "G4HLIW32.DLL" (ByVal Filename As String, ByVal VarName As String, ByRef lpPCPObj As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_LoadObjectByIndex Lib "G4HLIW32.DLL" (ByVal Filename As String, ByVal Index As Integer, ByRef lpPCPObj As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_LoadObjectByEntry Lib "G4HLIW32.DLL" (ByVal Filename As String, ByVal Entry As Integer, ByRef lpPCPObj As T_HLI_PCP_RW) As Integer
Public Declare Function IBS_HLI_PCP_InitObject Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Segment As Integer, ByVal Position As Integer, ByVal Index As Integer, ByVal Subindex As Integer, ByVal POT As Integer, ByVal nElements As Integer, ByRef lpData As Any) As Integer
Public Declare Function IBS_HLI_PCPObj_SetInt16 Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Offset As Integer, ByVal VInt16 As Integer) As Integer
Public Declare Function IBS_HLI_PCPObj_SetInt32 Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Offset As Integer, ByVal VInt32 As Long) As Integer
Public Declare Function IBS_HLI_PCPObj_SetFloat Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Offset As Integer, ByVal VFloat As Single) As Integer
Public Declare Function IBS_HLI_PCPObj_SetString Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Offset As Integer, ByVal lpS As String) As Integer

```

```

Public Declare Function IBS_HLI_PCPObj_GetInt16 Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Offset As Integer, ByRef lpInt16 As Integer) As Integer
Public Declare Function IBS_HLI_PCPObj_GetInt32 Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Offset As Integer, ByRef lpInt32 As Long) As Integer
Public Declare Function IBS_HLI_PCPObj_GetFloat Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Offset As Integer, ByRef lpFloat As Single) As Integer
Public Declare Function IBS_HLI_PCPObj_GetString Lib "G4HLIW32.DLL" (ByRef lpPCPObj As T_HLI_PCP_RW, ByVal Offset As Integer, ByVal lpS As String) As Integer

```

' Controller-Services

```

Public Declare Function IBS_HLI_ActivateWatchdog Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal WDT As Integer) As Integer

```

' INTERBUS-Services

```

Public Declare Function IBS_HLI_StartBus Lib "G4HLIW32.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_StopBus Lib "G4HLIW32.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_AlarmStop Lib "G4HLIW32.DLL" (ByVal CID As Integer) As Integer
Public Declare Function IBS_HLI_SwitchRemoteBus Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal segment_nr As Integer, ByVal switch_code As Integer) As Integer
Public Declare Function IBS_HLI_SwitchLocalBus Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal segment_nr As Integer, ByVal switch_code As Integer) As Integer
Public Declare Function IBS_HLI_SwitchGroup Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal group_nr As Integer, ByVal Alternative As Integer, ByVal switch_code As Integer) As Integer
Public Declare Function IBS_HLI_ChangeDeviceState Lib "G4HLIW32.DLL" (ByVal CID As Integer, ByVal Segment As Integer, ByVal Position As Integer, ByVal new_state As Integer) As Integer

```

' COP communication functions

```

Public Declare Function IBS_HLI_SendMessageToCop Lib "G4HLIW32.DLL" (ByVal COPID As Integer, ByRef Msg As T_HLI_MSG, timeout As Integer) As Integer
Public Declare Function IBS_HLI_ReceiveMessageFromCop Lib "G4HLIW32.DLL" (ByVal COPID As Integer, ByRef Msg As T_HLI_MSG) As Integer
Public Declare Function IBS_HLI_WaitForMessageFromCop Lib "G4HLIW32.DLL" (ByVal COPID As Integer, ByRef Msg As T_HLI_MSG, ByVal timeout As Integer) As Integer

```

```

Public Declare Function IBS_HLI_ReadDataFromCop Lib "G4HLIW32.DLL" (ByVal COPID As Integer, ByVal from_offset As Integer, ByVal nbytes As Integer, ByRef destination As Any) As Integer
Public Declare Function IBS_HLI_WriteDataToCop Lib "G4HLIW32.DLL" (ByVal COPID As Integer, ByVal to_offset As Integer, ByVal nbytes As Integer, ByRef source As Any) As Integer

```

```

Public Declare Function IBS_HLI_GetCopDiagnostic Lib "G4HLIW32.DLL" (ByVal COPID As Integer, ByRef StateReg As Integer, ByRef ParameterReg As Integer) As Integer

```

#End If

```

' -----**
'               END OF FILE               **
' -----**

```

## **Anexo C – Códigos-fonte dos programas dos MD's em IPos Plus®**

O conteúdo deste anexo é a versão do programa de controle com última atualização em \_\_\_\_\_. Autoria: Willians Anderson.

Listagem do programa IPOS PLUS

Não Disponível

## **Anexo D – Tabelas de Descrição do Projeto de Fiação**

O conteúdo deste anexo é referente à fiação da máquina, tal qual está construída. Autoria: Daniel T. Barros Moraes e Marcelo Boczeko.

Este subsistema tem a finalidade de interfacear os diversos subsistemas e prover dados aos dois sistemas de controle. A fiação foi desenvolvida em paralelo com a programação do equipamento, já que esta determinava e dependia das definições de bornes virtuais (chaves virtuais que simbolizavam para os programas, o estado dos diversos sensores, sinais de comunicação e chaves físicas implementadas). A fiação compões-se de oito módulos básicos:

- Potência;
- Interbus;
- Encoders;
- Proteção Térmica;
- Interface de Comunicação, Programação e Configuração;
- Sensoreamento;
- Interface com chaves e sinais;
- Joystick.

## 1 - Potência

A fiação de potência compõe-se de uma alimentação trifásica de 380 V, um transformador 380 V - 220 V trifásico, com capacidade de 8 kVA, um disjuntor de três fases e um neutro, fiação do disjuntor até os Conversores de Frequência, Resistores de Frenagem e alimentação dos motores (desde os conversores de frequência). O conector de potência que liga o gabinete à mesa possui pinagem discriminada na Tabela 1.

**Tabela 1 – Pinagem do Conector de Potência.**

| Pino | Sinal  | Motor |
|------|--------|-------|
| 1    | U      | 1     |
| 2    | V      | 1     |
| 3    | W      | 1     |
| 4    | Neutro | 1     |
| 5    | U      | 2     |
| 6    | V      | 2     |
| 7    | W      | 2     |
| 8    | Neutro | 2     |
| 9    | U      | 3     |
| 10   | V      | 3     |
| 11   | W      | 3     |
| 12   | Neutro | 3     |
| 13   | U      | 4     |
| 14   | V      | 4     |
| 15   | W      | 4     |
| 16   | Neutro | 4     |

## 2 - Interbus

A comunicação Interbus se dá desde o computador até o MoviDrive 4. Deste, parte outro cabo até o MoviDrive3, deste para o 2 e do 2 para o 1. O cabeamento é feito especificamente para a comunicação Interbus (padrão Remote Bus) e possui 5 fios e malha de aterramento. Os cabos que ligam os MoviDrive's não precisam ser compridos (têm 0,5 m, aproximadamente), mas o que liga o computador ao MoviDrive 4, para efeito de manutenibilidade, é construído com um comprimento de aproximadamente 10 metros. Cada cabo é terminado nas pontas com conectores DB-9 blindados. Numa extremidade, um conector fêmea; na outra, um conector macho. A pinagem da fiação Interbus foi realizada conforme Tabela 2 e a sequência de ligação de rede Interbus **deve** seguir a Tabela 3.

**Tabela 2 – Pinagem dos cabos Interbus**

| Pino | Cor Fio |
|------|---------|
| 1    | AM      |
| 2    | CZ      |
| 3    | MR      |
| 4    | -       |
| 5    | -       |
| 6    | VD      |
| 7    | PZ      |
| 8    | -       |
| 9    | -       |

**Tabela 3 – Sequência de ligações de Interbus**

| Fêmea   | Macho |
|---------|-------|
| Comput. | MD 4  |
| MD 4    | MD 3  |
| MD 3    | MD 2  |
| MD 2    | MD 1  |

### 3 Encoders

A comunicação entre MD's e os encoders de seus respectivos motores se dá através de cabos de 10 vias, com carcaça aterrada. De cada MD, parte um cabo com conector DB-9 macho e segue internamente ao armário, até terminar numa conexão DB-9 fêmea. De lá parte outro cabo, conectado através de conector DB-9 macho até o encoder do respectivo motor. A pinagem está apresentada na Tabela 4.

**Tabela 4 – Pinagem dos cabos dos encoders**

| Pino | Cor Fio | Sinal    | Encoder   |
|------|---------|----------|-----------|
| 1    | LA      | A        | K1        |
| 2    | VM      | B        | K2        |
| 3    | MR      | C        | K0        |
| 4    | -       | -        | -         |
| 5    | BR      | GND      | terra     |
| 6    | VD      | <u>A</u> | <u>K1</u> |
| 7    | AZ      | <u>B</u> | <u>K2</u> |
| 8    | CZ      | <u>C</u> | <u>K0</u> |
| 9    | BR/AZ   | .+VCC    | UB        |

#### 4 Proteção Térmica

A proteção térmica dos motores é implementada através da ligação de dois fios do MD com dois fios do respectivo motor. Estes fios seguem por um cabo de 10 vias desde o MD até o interior do gabinete, onde são desviados para um cabo de 8 vias que comporta todos os sinais de proteção térmica juntos. O cabo termina em um conector DB-9 fêmea. No exterior do gabinete, um outro cabo de 8 vias é conectado, através de conector DB-9 macho. A outra extremidade é ligada diretamente aos fios de proteção térmica dos motores.

**Tabela 5 – Pinagem do cabo de Proteção Térmica.**

| Pino DB-9 | Sinal   | MD | Pino MD    | Fio |
|-----------|---------|----|------------|-----|
| 1         | Motor 1 | 1  | X . 10 . 1 | AM  |
| 2         | Ret. M1 | 1  | X . 10 . 2 | LA  |
| 3         | -       | -  | -          | -   |
| 4         | Motor 2 | 2  | X . 10 . 1 | VM  |
| 5         | Ret. M2 | 2  | X . 10 . 2 | MR  |
| 6         | Motor 3 | 3  | X . 10 . 1 | AZ  |
| 7         | Ret. M3 | 3  | X . 10 . 2 | VD  |
| 8         | Motor 4 | 4  | X . 10 . 1 | CZ  |
| 9         | Ret. M4 | 4  | X . 10 . 2 | PT  |

## 5 Interface de Comunicação, Programação e Configuração

A comunicação entre o computador e os MD's é feita através do cabeamento Interbus. Entretanto, durante a programação e configuração dos MD's é necessária uma comunicação extra. Esta comunicação é feita através de um cabo de 8 vias, com conectores DB-9 macho e fêmea em suas extremidades. Este cabo é ligado a este computador. O final do cabo é acoplado a um conector DB-9 fêmea na base do gabinete. De lá é trazido um cabo de 8 vias, para comunicação com o Conversor de Interface RS-232 / RS-485. O acoplamento do cabo com o conversor é feito por conector DB-9 macho.

**Tabela 6 – Pinagem do cabo de comunicação RS232.**

| Pino - DB-9 | Sinal | Cor |
|-------------|-------|-----|
| 1           | -     | -   |
| 2           | RxD   | MR  |
| 3           | TxD   | VD  |
| 4           | -     | -   |
| 5           | GND   | AZ  |
| 6           | -     | -   |
| 7           | -     | -   |
| 8           | -     | -   |
| 9           | -     | -   |

**Tabela 7 – Pinagem do cabo de comunicação RS485.**

| Conector | fio | MD  | Contato     | Sinal    |
|----------|-----|-----|-------------|----------|
| 1        | VM  | 4   | X . 13 . 8  | . +24V   |
| 2        | PT  | 4   | X . 13 . 9  | . -24V   |
| 3        | MR  | All | X . 13 . 10 | RS 485 + |
| 4        | AZ  | All | X . 13 . 11 | RS 485 - |
| 5        | -   |     | -           | -        |

## 6 Sensoreamento

A comunicação entre os MD's e os diversos sensores é feita através dos cabos de 8 vias que partem dos MD's. No interior do equipamento, estes cabos unem-se e os fios seguem agrupados em um cabo de 26 vias, até terminarem em conector DB-25 fêmea. Do lado externo é conectado, através de conector DB-25 macho, um cabo de 26 vias que leva os fios até a mesa, onde estão instalados os sensores. Os sinais são separados e os fios dos sensores de fim de curso são diretamente ligados nos fios do cabo de 26 vias. Os fios que vêm da plataforma são agrupados num cabo de 14 vias e, no momento da conexão com o cabo de 26 vias, esta é feita através de conectores DB-15 (macho nos fios dos sensores e fêmea nos fios do cabo 26 vias).

**Tabela 8 – Pinagem e fiação dos sensores.**

| Sinal | Sensor | Fio - Sensor | Fio 14 vias | Fio 26 vias | Pino DB-25 | Fio Interno | Fio MD | MD | Pino MD    |
|-------|--------|--------------|-------------|-------------|------------|-------------|--------|----|------------|
| +     | SP     | marrom       | AM          | BR / AZ     | 1          | BR / AZ     | AM     | 1  | X . 13 . 8 |
| s     | SP     | preto        | LA          | BR / AM     | 2          | BR / AM     | LA     | 1  | X . 13 . 2 |
| -     | SP     | azul         | VM          | BR / LA     | 3          | BR / LA     | VM     | 1  | X . 13 . 9 |
| +     | SG     | marrom       | AM          | BR / VD     | 4          | BR / VD     | MR     | 1  | X . 13 . 8 |
| s     | SG     | preto        | LA          | BR / MR     | 5          | BR / MR     | AZ     | 1  | X . 13 . 4 |
| -     | SG     | azul         | VM          | BR / CZ     | 6          | BR / CZ     | VD     | 1  | X . 13 . 9 |
| 7     |        |              |             |             |            |             |        |    |            |
| +     | CB     | marrom       | AZ          | AZ / BR     | 8          | AZ / BR     | AM     | 2  | X . 13 . 8 |
| s     | CB     | preto        | VD          | AZ / AM     | 9          | AZ / AM     | LA     | 2  | X . 13 . 2 |
| -     | CB     | azul         | VI          | AZ / LA     | 10         | AZ / LA     | VM     | 2  | X . 13 . 9 |
| +     | BP     | marrom       | BR          | AZ / VD     | 11         | AZ / VD     | MR     | 2  | X . 13 . 8 |
| s     | BP     | preto        | CZ          | AZ / MR     | 12         | AZ / MR     | AZ     | 2  | X . 13 . 3 |
| -     | BP     | azul         | PT          | AZ / CZ     | 13         | AZ / CZ     | VD     | 2  | X . 13 . 9 |
| +     | FC1    | marrom       |             | AM / BR     | 14         | AM / BR     | CZ     | 1  | X . 13 . 8 |
| s     | FC1    | azul         |             | AM / AZ     | 15         | AM / AZ     | VI     | 1  | X . 13 . 5 |
| 16    |        |              |             |             |            |             |        |    |            |
| +     | FC2    | marrom       |             | AM / LA     | 17         | AM / LA     | CZ     | 2  | X . 13 . 8 |
| s     | FC2    | azul         |             | AM / VD     | 18         | AM / VD     | VI     | 2  | X . 13 . 4 |
| 19    |        |              |             |             |            |             |        |    |            |
| 20    |        |              |             |             |            |             |        |    |            |
| +     | FC3    | marrom       |             | AM / MR     | 21         | AM / MR     | CZ     | 3  | X . 13 . 8 |
| s     | FC3    | azul         |             | AM / CZ     | 22         | AM / CZ     | VI     | 3  | X . 13 . 2 |
| 23    |        |              |             |             |            |             |        |    |            |
| +     | FC4    | marrom       |             | AM / VI     | 24         | AM / VI     | CZ     | 4  | X . 13 . 8 |
| s     | FC4    | azul         |             | BR / VI     | 25         | BR / VI     | VI     | 4  | X . 13 . 2 |

## **7 Interface com chaves e sinais**

A interface entre o equipamento é feita através de bornes virtuais dos MD's. Os fios que conduzem os sinais de chaves, sinais lógicos da estrutura de mídia e os fios de alimentação e de comunicação com o conversor de interfaces são conduzidos através de cabos de 10 vias. No interior do armário estes cabos são desmembrados e as ligações necessárias feitas. No caso de sinais e de chaves, é utilizado um cabo de 8 vias para dar liberdade de atuação na implementação da estrutura de mídia.

### 3.5.8 Joystick

A fiação do joystick é praticamente apenas uma extensão, para dar a liberdade de sua colocação a uma distância razoável. Como o sinal de saída do joystick é analógico, o comprimento do fio influi na qualidade do sinal, mas não há erros relevantes por causa da inserção de ruídos. Compõe-se de um cabo interno ao armário de cerca de 1 metro de comprimento, com um conector DB-15 macho numa ponta e um DB-15 fêmea na ponta que fica no exterior do armário. Fora do armário, um outro cabo, de cerca de 5 metros, com um conector DB-15 macho em uma ponta e um conector DB-15 fêmea na outra, serve como extensão externa para o joystick.

**Tabela 9 – Pinagem do cabo do joystick.**

| DB-15 | Cor |
|-------|-----|
| 1     | VD  |
| 2     | BR  |
| 3     | LA  |
| 4     | BR  |
| 5     | -   |
| 6     | MR  |
| 7     | AZ  |
| 8     | -   |
| 9     | BR  |
| 10    | -   |
| 11    | -   |
| 12    | -   |
| 13    | -   |
| 14    | CZ  |
| 15    | -   |

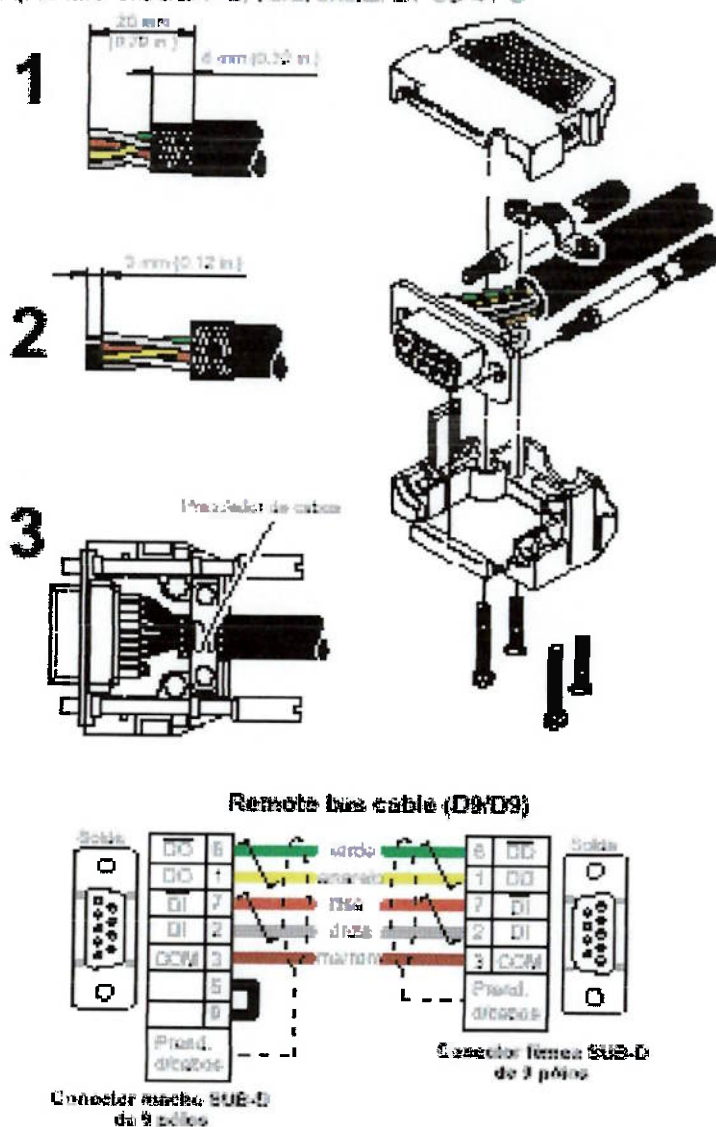
## Anexo E – Indicações de Montagem para Remote Field Bus Cable

O conteúdo deste anexo mostra a maneira de se construir o cabo de conexão remota do padrão Interbus.

## Indicações de montagem para o remote bus cable

**Instruções de montagem:** conectores macho e fêmea D-SUB de 8 pins, soldáveis

Tipo: 159 (SUB ML) referência: 27 58 47 3

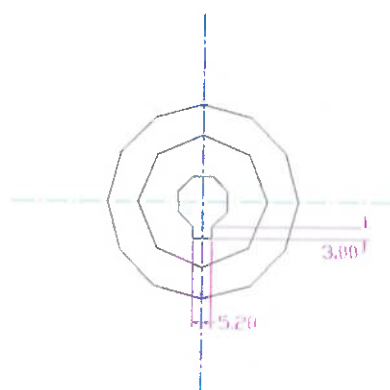
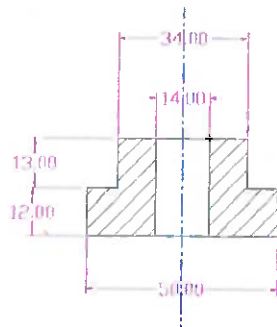


**Figura 7 Remote bus cable com conexão soldada**

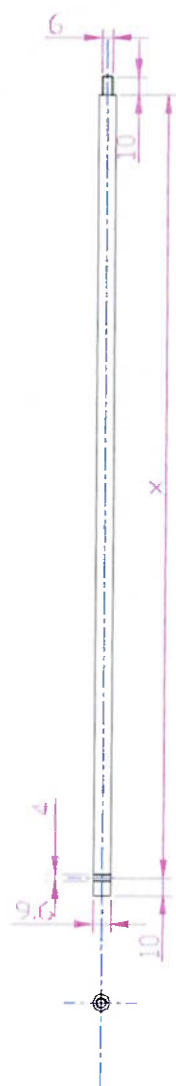
## **Anexo F – Desenhos de Fabricação**

O conteúdo deste anexo apresenta alguns desenhos de fabricação relativos às peças construídas para emprego nesta máquina. É de Autoria de Daniel T. B. Moraes e de Marcelo Boczko.

As escalas não devem ser consideradas.

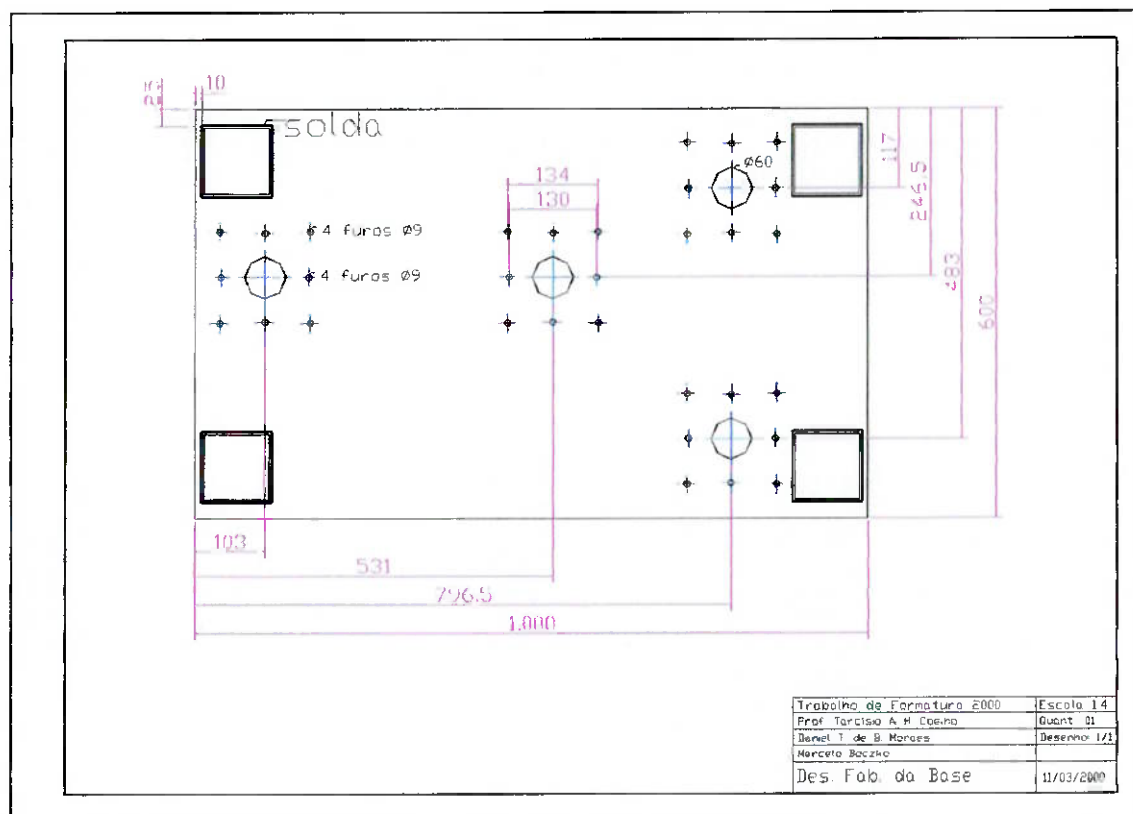


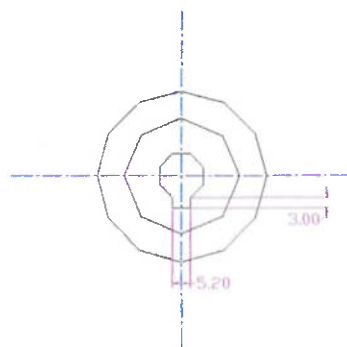
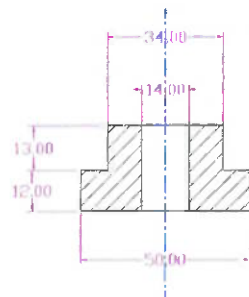
|                             |             |
|-----------------------------|-------------|
| Trabalho de Formatura 2000  | Escala 1:1  |
| Prof. Tercisio A. H. Coelho | Quant. 04   |
| Daniel T. de B. Moraes      | Desenho 1/1 |
| Marcelo Boczko              |             |
| Acoplamento do Motor        | 11/03/2000  |



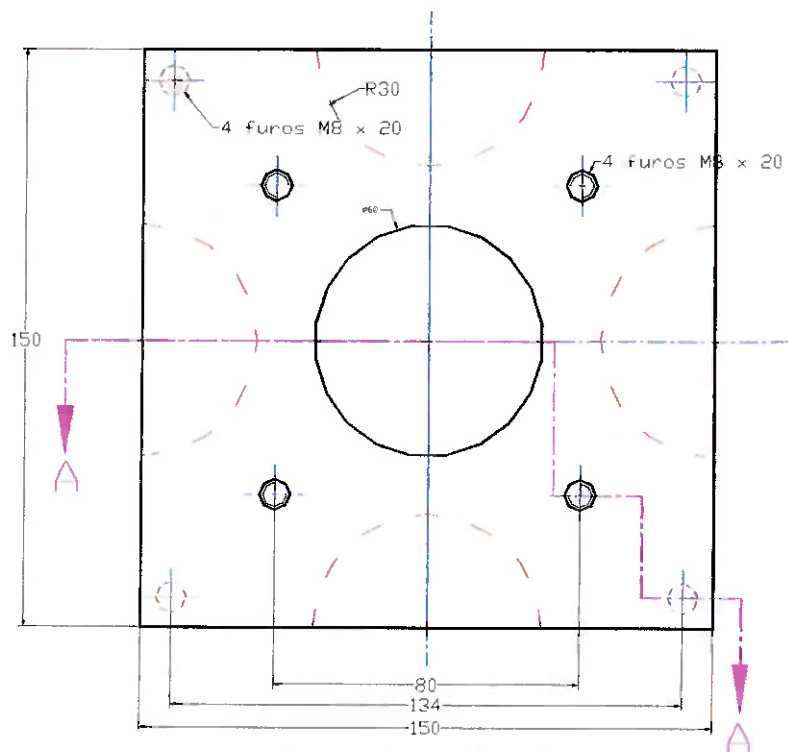
Observações: Barras cilíndricas de Alumínio, diâmetro de 10 a 15 mm  
comprimentos  $x = 440$ ,  $x = 480$ ,  $x = 480$ ,  $x = 500$  mm

|                             |             |
|-----------------------------|-------------|
| Trabalho de Formatura 2000  | Escala 1:1  |
| Prof. Tarcísio A. H. Coelho | Quant. 04   |
| Daniel T. de B. Moraes      | Desenho 1/1 |
| Marcelo Boczeko             |             |
| Acoplamento do Motor        | 11/03/2000  |

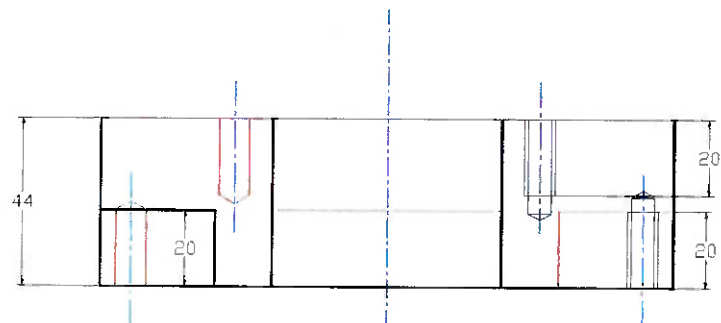




|                             |             |
|-----------------------------|-------------|
| Trabalho de Formatura 2000  | Escala 1:1  |
| Prof. Tarcísio A. H. Coelho | Quant. 04   |
| Daniel T. de B. Moraes      | Desenho 1/1 |
| Marcelo Boczek              |             |
| Acoplamento do Motor        | 11/03/2000  |

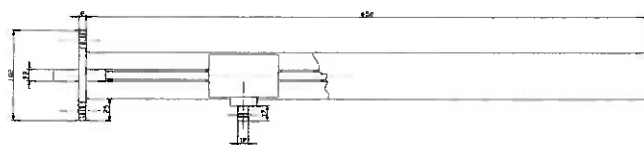


Corte A-A

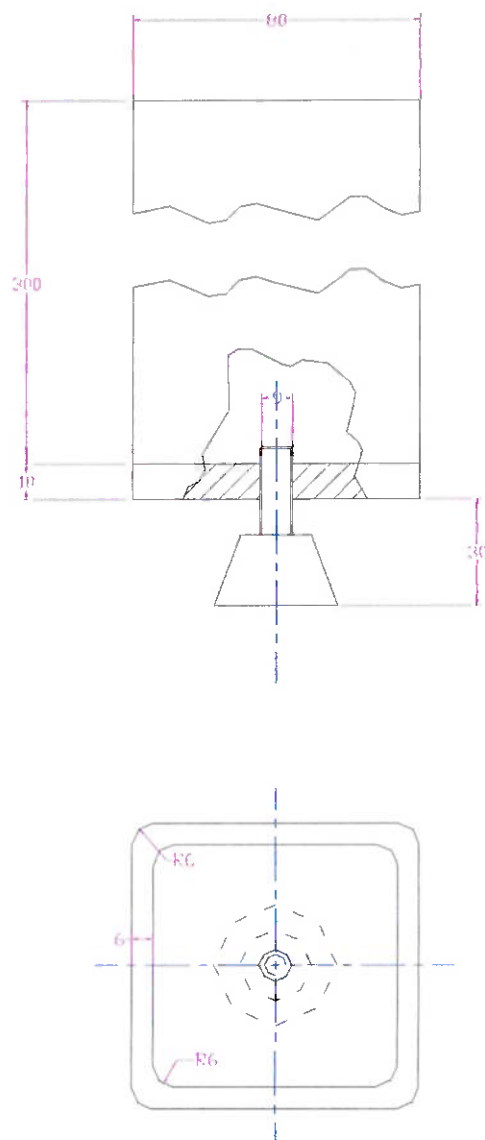


material: alumínio

|                             |             |
|-----------------------------|-------------|
| Trabalho de Formatura 2000  | Escala 1:1  |
| Prof. Tarcísio A. H. Coelho | Quant. 04   |
| Daniel T. de B. Moraes      | Desenho 1/1 |
| Marcelo Boczko              |             |
| Des. Fab. do Espaçador      | 11/03/2000  |



|                             |             |
|-----------------------------|-------------|
| Trabalho de Formatura 2000  | Escola 1:2  |
| Prof. Torcísio A. H. Coelho | Quant. 01   |
| Daniel T. de E. Moraes      | Desenho 1/1 |
| Marcelo Boczeko             |             |
| Haste de Fuso               | 11/03/2000  |



|                             |             |
|-----------------------------|-------------|
| Trabalho de Formatura 2000  | Escala 1:1  |
| Prof. Tarcísio A. H. Coelho | Quant: 04   |
| Daniel T. de B. Moraes      | Desenho 1/1 |
| Marcelo Baczko              |             |
| Pês                         | 11/03/2000  |

## Anexo G – FieldBus: Visão Geral

O conteúdo deste anexo explicita algumas características do sistema de comunicação de dados FieldBus. É de Autoria de Daniel T. B. Moraes, Marcelo Boczko e Roberto Sakura.

### 1 – FieldBus

O FieldBus é uma rede de transmissão de dados de sistema aberto para comunicação com equipamentos de instrumentação e controle de plantas industriais, tais como transmissores, atuadores e controladores, podendo, inclusive, ser utilizado em aplicações que requeiram especificações quanto aos requisitos de segurança intrínseca. Esta rede é do tipo digital, serial, *half-duplex* (comunicação bidirecional, porém, em uma única direção a cada instante) e *multidrop* (comunicação entre vários equipamentos conectados à rede).

O meio físico é implementado através de fios de par trançado conectando um cartão padrão ISA (interior ao computador) diretamente ao aos instrumentos/atuadores ou aos transmissores FieldBus. Na rede FieldBus, podem ser conectados equipamentos tipo *bus powered* (alimentados através da própria rede) ou com alimentação independente, permitindo, no primeiro caso, alimentação e comunicação em apenas um par de fios (ou cabo coaxial), minimizando os custos de implementação do sistema de controle e aquisição de dados.

A alimentação dos dispositivos ligados ao FieldBus pode ser feita através dos fios que compõem a rede ou por alimentação independente, sendo permitida a redundância e também um sistema misto de alimentação. As fontes que alimentam os componentes da rede são fornecidas de acordo com as especificações do FieldBus.

O sistema FieldBus consiste basicamente de três níveis: o nível de usuário, o nível de controle e aquisição e o nível de chão de fábrica.

O nível de usuário é implementado através de uma interface computacional, que se utiliza de um Sistema Supervisório e de um Sistema de Controle e Aquisição de Dados, capaz de implementar funções de controle via *software*.

No nível de controle, é implementado o controle que será aplicado ao processo, através de funções de controle. As funções de controle são criadas através da união de

funções de controle pré-definidas, como por exemplo controle proporcional-integral, controle por operações aritméticas e outros. A implementação destas funções de controle é feita na placa PCI, conectada ao computador mas, para o usuário, é como se esta fosse implementada diretamente no sensor ou no atuador ou em qualquer ponto da planta que se queira e trabalhasse diretamente com os elementos utilizados para projeto do controle.

No nível de chão de fábrica são acoplados os transmissores FieldBus e nestes os atuadores e os sensores da planta.

O FieldBus foi estruturado com o objetivo de eliminar as conversões A/D, minimizar interferências e erros decorrentes de comunicação até então existentes. Além disso, o sistema permite múltiplos computadores gerenciando e monitorando a planta (conectados a partir de uma rede EtherNet sob protocolo TCP/IP), ocasionando maior facilidade de operação, pois pode ser operado em plataformas conhecidas pela maioria dos usuários como Windows e Windows NT. O sistema permite ainda flexibilidade e interoperabilidade entre sistemas (como DDC e SDCD), facilitando a transição para o FieldBus que é um Sistema de Controle de Campo – FCS com seus transmissores inteligentes, minimizando assim o custo de implementação. A disposição da rede permite ainda fácil expansão devido à possibilidade de se conectar até 32 dispositivos FieldBus em cada canal, permitindo a conexão de até 128 instrumentos/atuadores para cada placa PCI (Interface de Controle de Processo – *Process Control Interface*). Pode ainda ser utilizada placa BC, que permite controle de apenas um canal

Devido ao protocolo de comunicação interno da rede FieldBus, o fluxo de informações ocorre em tempo real, garantindo que a variável sendo transmitida está atualizada. Isto é feito através de um sistema de pedidos (*tokens*) e um gerenciador de tarefas. Sendo assim, o gerenciador administra quem transmite o dado e quem recebe o dado a cada instante.

A malha de controle consiste na união de blocos funcionais contidos dentro dos transmissores inteligentes com a possibilidade de se anexar os blocos não pertencentes aos transmissores, sendo a configuração realizada a partir de parâmetros guardados na placa PCI.

## **2 – Protocolo de Comunicação FieldBus**

A comunicação entre os diversos dispositivos da rede FieldBus é controlada pela placa **PCI 302** (ou **BC1**). Ela é feita através do mesmo par de fios trançados utilizados para alimentação dos dispositivos ligados à rede FieldBus.

A comunicação pode ser feita em baixa velocidade (31,25 Kbps), permitindo alimentação pelo barramento ou em alta velocidade (1 Mbps ou 2,5 Mbps), sem alimentação por barramento.

Quando é utilizada a comunicação tipo H1, ou seja, baixa velocidade de transmissão, a alimentação do FieldBus pode ser configurada para operar com corrente constante ou com tensão constante. Caso a configuração seja de tensão constante, a comunicação ocorre com pulsos trapezoidais de tensão. No caso do corrente constante, os pulsos são de corrente.

O nível de enlace é responsável pelo endereçamento correto dos blocos de informação e pela verificação da integridade, além de outras informações de comunicação, como o endereço do emissor.

## **3 – Transmissores de FieldBus**

Transmissores são os equipamentos conectados aos sensores e atuadores, convertendo o sinal analógico ou digital recebido e retransmitindo-o de acordo com o protocolo FieldBus, em valores percentuais dos valores limites configurados. Algumas funções, como AI e AO são implementadas nestes dispositivos. Existem diversos dispositivos, cada um adequado a um tipo de grandeza a ser mensurada ou aplicada. São alguns deles:

- **PSI 302** – Filtro de linha para a Fonte de Alimentação;
- **TT 302** – Transmissor FieldBus de Temperatura;
- **LD 302** – Transmissor FieldBus de Pressão;

- **FP 302** – Conversor de FieldBus para Pressão;
- **FI 302** – Conversor de Corrente para FieldBus;
- **IF 302** – Conversor de FieldBus para Corrente;
- **FP 302** – Conversor Pneumático para FieldBus;
- **FY 302** – Posicionador FieldBus;
- **PSI 302** – Impedância para Fonte de Alimentação FieldBus.

Além disto, a fonte que alimenta a rede é um sistema de *hardware* independente, com fonte e terminação de linha. A fonte utilizada pela Smar (fabricante de FieldBus e componentes) é a **PS 302**. O terminador de Linha é o **BT 302**.

#### 4 – Funções de Controle Pré-Definidas

Diversas são as funções de controle pré-definidas no sistema FieldBus, que podem ser unidas como blocos, gerando a função de controle que se deseja implementar na planta. As principais são:

- **AI – Entrada Analógica** – O bloco AI obtém os dados do transdutor de entrada (sensor) e deixa-o disponível para outros blocos funcionais na sua saída. Esta saída é transmitida na forma de um valor percentual;
- **AO – Saída Analógica** – O bloco AO é um bloco funcional utilizado por equipamentos que trabalham como elementos de saída em um loop de controle, como válvulas, atuadores, posicionadores, etc. Ele recebe o sinal de outro bloco funcional e o converte para um sinal compatível com a necessidade do *hardware*;
- **PID – Controle Proporcional-Integral-Derivativo** – O bloco PID é a chave para vários diagramas de controle e é usado quase que universalmente com exceção do PD, que é usado quando o processo faz a própria integração. Enquanto existir um erro, a função PID integrará o erro que movendo a saída na direção a corrigir o erro.

Blocos PID podem ser ligados em cascata. O bloco PID trabalha bem para constantes de tempo relativamente pequenas, encontradas na primeira linha de controle de processo;

- **ARTH – Controle Aritmético** – O Bloco Aritmético somente opera sobre entradas escalares. É usado para calcular uma saída escalar, como uma função de até quatro saídas das cinco entradas de acordo com um algoritmo selecionado. A determinação de qual das operações pré definidas será realizada é feita a partir de diversos algoritmos que estão embutidos e podem ser selecionados pelo operador. O operador tem seis constantes que ele pode escrever em qualquer modo para obter uma saída desejada. Entradas não conectadas não são consideradas. Somente as constantes correspondentes terão efeito. O polinômio de quarta ordem pode ser usado para implementar as séries de Taylor (Maclaurin), obtendo um valor aproximado de por exemplo, funções logarítmicas e trigonométricas.;
- **INT – Integrador** – O bloco funcional integrador integra uma variável na função de tempo ou acumula a contagem de um bloco de entrada de pulso. O valor acumulado ou integrado é comparado ao pre-trip e aos limites trip, gerando sinais discretos quando esses limites são alcançados. O valor integrado pode aumentar, partindo do zero, ou diminuir partindo do valor trip (parâmetro SP). Para determinar o total de leituras incertas ou ruins, o bloco integra as variáveis com status bons ou bons e incertos separadamente (parâmetro RTOTAL);
- **ISS – Seletor de Sinal de Entrada** – O bloco seletor de sinal fornece seleção de até três entradas e era uma saída baseada na ação configurada. Este bloco normalmente receberá sua entrada de um bloco AI ou de outro bloco. Em adição à seleção de sinal o bloco também pode selecionar o maior, o menor, o intermediário, o multiplex ou a primeira seleção boa.

## 5 – Sistema Supervisório e Sistema de Configuração

Os Sistemas Supervisório e de Configuração são os *softwares* responsáveis pela interface do usuário com o sistema físico e sua arquitetura e com as funções de controle a serem implementadas.

O Sistema de Configuração (implementado pelo *software Syscon*) é responsável pela configuração do controle da planta, sendo realizado por pessoal especializado no sistema de controle. O controle do processo inicia-se com a alocação dos endereços e identificação dos componentes lógicos do FieldBus (Blocos Funcionais), implementados pelos dispositivos físicos presentes na rede. O Sistema de Configuração detecta automaticamente os dispositivos pendurados à rede e recebe deles as informações sobre os blocos de controle correspondentes embutidos. Estes blocos ficam a disposição do usuário (implementador do sistema de controle), que dá a eles nomes adequados (*tags*) para posterior conexão (“linkagem”).

O processo segue então com a “linkagem” dos blocos existentes com outros blocos, disponíveis por *software* e implementados na placa PCI (ou BC1). Alguns atributos dos blocos são determinados e configurados já neste processo, enquanto outros, alteráveis durante a operação da planta, serão controlados pelo Sistema Supervisório.

O Sistema Supervisório (implementado pelo *software AIMAX*) é responsável para monitoração e operação da planta. Nele podem ser inseridos pelo usuário (operador da planta) os valores de variáveis de processo adequados para o funcionamento da planta. O AIMAX recebe e passa os valores adequados aos parâmetros do *software* de configuração através dos *tags* neste definidos.

A conexão dos Sistemas de Configuração e Sistema Supervisório é aberta, ou seja, os *softwares* utilizados não são necessariamente os apresentados, sendo compatíveis com diversos outros.