

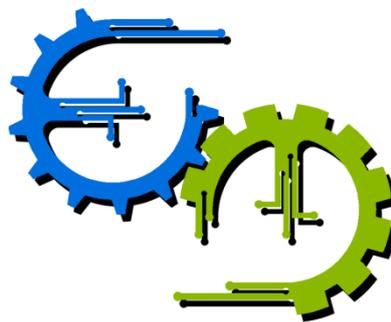
TRABALHO DE GRADUAÇÃO

**INSTALAÇÃO, CONFIGURAÇÃO E
INTEGRAÇÃO DOS DISPOSITIVOS DE
CONTROLE DE UM ROBÔ CARTESIANO**

Por,

Pedro Leoncio da Silva Neto

Brasília, julho de 2018



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

INSTALAÇÃO, CONFIGURAÇÃO E INTEGRAÇÃO DOS DISPOSITIVOS DE CONTROLE DE UM ROBÔ CARTESIANO

POR,

Pedro Leoncio da Silva Neto

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Guilherme Caribé de Carvalho, UnB/ ENM

(Orientador)

Prof. Carlos Humberto Llanos Quintero,

UnB/ENM

Prof. Walter de Britto Vidal Filho, UnB/ENM

Brasília, julho de 2018

FICHA CATALOGRÁFICA

LEONCIO DA SILVA NETO, PEDRO

Instalação, configuração e integração dos dispositivos de controle de um robô cartesiano,
[Distrito Federal] 2018.

xiii, 92 p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, Ano). Trabalho de Graduação –
Universidade de Brasília, Faculdade de Tecnologia.

- | | |
|-------------|--------------|
| 1. Robótica | 2. Automação |
| 3. Controle | 4. Projeto |

I. Mecatrônica/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

LEONCIO DA SILVA NETO, P. (2018). Instalação, configuração e integração dos dispositivos de controle de um robô cartesiano. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 06, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 92 p.

CESSÃO DE DIREITOS

AUTOR: Pedro Leoncio da Silva Neto.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Instalação, configuração e integração dos dispositivos de controle de um robô cartesiano.

GRAU: Engenheiro

ANO: 2018

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Pedro Leoncio da Silva Neto

Grupo de Automação e Controle (GRACO)

Faculdade de Tecnologia

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 – Brasília – DF – Brasil

AGRADECIMENTOS

Agradeço a cima de tudo aos meus pais, Angela Maria e Pedro Leoncio, pois sem o apoio incondicional deles jamais seria possível realizar meu sonho de me tornar engenheiro. Apesar de todos os problemas que ocorreram ao longo de nossas vidas, a garra e a vontade deles em me verem formados jamais foi abalada e só nos deram mais forças para continuar lutando.

Agradeço também a todos os meus amigos que me acompanharam desde o início da minha jornada no curso de engenharia mecatrônica. Em especial ao Julio Francisco, por passar noites acordado me auxiliando no desenvolvimento e escrita deste trabalho, ao Lucas Neves, por não medir esforços em me ajudar sempre que fosse necessário caso surgisse alguma dúvida na programação, à Vanessa Lucena, por sempre conseguir tempo para me ensinar a usar softwares com o qual eu ainda não tinha trabalhado e ao Vinícius Henrique, pelo apoio moral e ajuda integral na escrita deste projeto. Sem essas pessoas, a realização deste trabalho não seria possível.

Por fim, agradeço ao professor Guilherme caribé, pois, apesar da demora em mostrar resultados, ele jamais desistiu de me orientar e sempre que possível encontrou tempo para me auxiliar em questões que apareciam e mostrar detalhes que me passaram despercebidos, e aos técnicos do GRACO, Pereira e Arthur, por estarem sempre dispostos a ajudar.

Pedro Leoncio da Silva Neto.

RESUMO

Robôs industriais têm se estabelecido na indústria manufatureira por mais de 30 anos e são utilizados das mais variadas formas, de modo que substituem o trabalho braçal em tarefas que demandem acurácia e trabalhos repetitivos. Dentro da indústria, o tipo de robô mais comumente utilizado é o manipulador, que realiza tarefas como empilhamento, pintura, ordenação e soldagem. Assim, desempenham atividades que seriam perigosas para o homem.

Tendo em vista o constante desenvolvimento da área de robótica industrial, investigações nessa temática são importantes para o desenvolvimento de novos projetos para quem tem interesse na área. Assim, baseado nos estudos bibliográficos e pesquisas relacionadas à área de robótica, foi desenvolvido o presente projeto, cujo objetivo foi realizar a instalação de um robô cartesiano, bem como a configuração, integração e instalação dos dispositivos de controle desse robô.

Nesse projeto são apresentadas as etapas de instalação, configuração e integração dos dispositivos de controle do robô cartesiano MAXR23-S42-H42-C42, da Schneider Electric. Seus dispositivos de controle são três servo-drives, utilizados para acionar os servomotores responsáveis pela movimentação dos eixos, e um controlador de movimentos, o LMC058LF42 da Schneider Electric, que é programado e envia os comandos aos servo-drives. Os dispositivos são instalados em um painel e este painel é alimentado pela rede geral de distribuição de energia do GRACO.

A etapa de instalação dos dispositivos foi baseada, principalmente, no manual da ABNT para instalação de dispositivos de baixa tensão NBR5410, e nos manuais de instalação dos dispositivos. A configuração e integração teve como base os guias do usuário dos elementos e os estudos realizados sobre o protocolo de rede CANopen – esse protocolo é utilizado para fazer a comunicação entre o controlador e os servo-drives. Por fim, foi realizado um programa para fazer a movimentação do robô e verificar se as etapas descritas foram feitas de maneira correta. Os procedimentos anteriormente descritos e os resultados obtidos se mostraram satisfatórios, visto que resultaram em um sistema robótico funcional, pronto para receber coordenadas e se movimentar de acordo com as trajetórias desejadas.

ABSTRACT

Industrial robots have been established in the manufacturing industry for more than 30 years and are used in a variety of ways, so that they replace manual labor in tasks that require accuracy and repetitive work. Within industry, the most commonly used type of robot is the manipulator, which performs tasks such as stacking, painting, sorting, and welding. Thus, they perform activities that would be dangerous to man.

In view of the constant development of the area of industrial robotics, investigations in this area are important for the development of new projects for those interested in the area. Thus, based on

bibliographic studies and researches related to the area of robotics, the present project was developed, whose objective was to carry out the installation of a Cartesian robot, as well as the configuration, integration and installation of control devices of this robot.

In this project are presented the steps of installation, configuration and integration of the devices of control of the cartesian robot MAXR23-S42-H42-C42, of Schneider Electric. Its control devices are three servo drives, used to activate the servo motors responsible for moving the axes, and a motion controller, the Schneider Electric's LMC058LF42, that is programmed and sends commands to the servo drives. The devices are installed in a panel and this panel is powered by GRACO's general power distribution network.

The installation stage of the devices was mainly based on the ABNT manual for the installation of low voltage devices NBR5410, and in the device installation manuals. Configuration and integration were based on the user guides of the elements and the studies carried out on the CANopen network protocol - this protocol is used to communicate between the controller and the servo drives. Finally, a program was carried out to make the movement of the robot and verify if the described steps were done correctly. The procedures described above and the results obtained were satisfactory, since the robotics system proved to be useful and ready to receive coordinates and to move accordingly to the desired trajectories.

SUMÁRIO

CAPÍTULO 1 – Introdução	1
1.1 Histórico e definições	1
1.2 Motivação	1
1.3 Objetivo	2
1.4 Estrutura do trabalho	2
CAPÍTULO 2 – Fundamentação Teórica	3
2.1 Robótica Industrial	3
2.1.1 Componentes de robôs	4
2.2 Tecnologias de controle, sensores, atuadores e drives de comando	7
2.2.1 Atuadores	7
2.2.2 Sensores	7
2.2.3 Tecnologias de controle	8
2.3 Protocolo de comunicação CANopen	9
2.4 Planejamento de trajetória	12
2.4.1 Planejamento de trajetórias por polinômio de terceira ordem	13
2.4.2 Planejamento de trajetórias por polinômio de quinta ordem	14
CAPÍTULO 3 – Especificação Técnica dos Dispositivos	15
3.1 Robô	15
3.2 Drives de acionamento	18
3.3 Controlador de movimentos	20
3.4 Somachine V4.1	22
3.5 Interface homem-máquina	24
3.6 Elementos complementares	25
CAPÍTULO 4 – Projeto de Instalação Elétrica	27
4.1 Alimentação do painel de controle e distribuição de energia para os dispositivos	28
4.2 Instalação elétrica dos Servo-drives	31
4.3 Instalação elétrica do Controlador de Movimentos	33

4.4 Instalação elétrica da Interface Homem-Máquina _____	35
4.5 Instalação elétrica dos servo-motores _____	36
4.6 Alimentação dos sensores de fim de curso e conexão com os servo-drives _____	41
CAPÍTULO 5 – Configuração dos Dispositivos e Programação do Controlador de Movimentos ____	43
5.1 Comissionamento dos drives e comunicação com o controlador de movimentos _____	43
5.2 Teste de locomoção dos eixos _____	48
5.3 Algoritmo _____	49
5.4 – Programa _____	50
CAPÍTULO 6 – Resultados _____	56
6.1 Instalação dos dispositivos no painel de controle _____	56
6.2 Programa-exemplo e teste de locomoção dos eixos _____	56
6.3 Programa final _____	56
6.3.1 Interface do usuário _____	56
6.3.2 Teste de reset dos erros _____	57
6.3.3 Teste de <i>homing</i> _____	57
6.3.4 Teste de posicionamento dos eixos _____	59
6.3.5 Teste do botão de parada _____	63
CAPÍTULO 7 – Conclusão _____	64
CAPÍTULO 8 – Considerações Finais _____	65
BIBLIOGRAFIA _____	66
Anexo I: Desenho técnico do robô _____	68
Anexo II: Característica dos atuadores _____	69
Anexo III: Configuração dos servo-motores _____	70
Apêndice I: Programa _____	71
Apêndice II: Desenho técnico da caixa de montagem _____	83
Apêndice III: Desenho técnico da instalação elétrica _____	86
Apêndice IV: Resultado da instalação do robô e dos dispositivos _____	88

LISTA DE FIGURAS

Figura 2.1	Manufatura robotizada para produção de automóveis.....	3
Figura 2.2	Exemplo de um manipulador linear.....	4
Figura 2.3	Efetuator terminal modelo garra articulada.....	4
Figura 2.4	Atuador linear.....	5
Figura 2.5	Encoder absoluto e seu disco codificado.....	5
Figura 2.6	Controlador de movimentos multi-eixos.....	5
Figura 2.7	Robô paralelo com cinco graus de liberdade.....	6
Figura 2.8	Arquitetura CAN padrão.....	9
Figura 2.9	CAN 2.0A.....	11
Figura 2.10	CAN 2.0B.....	11
Figura 3.1	Robô MAXR23-S42-H42-C42 da Schneider Electric.....	15
Figura 3.2	Eixo X.....	16
Figura 3.3	Eixo Y.....	16
Figura 3.4	Eixo Z.....	17
Figura 3.5	Família LXM da Schneider Electric.....	18
Figura 3.6	Visão geral do dispositivo LXM32M.....	20
Figura 3.7	Controlador de movimentos da família LMC058.....	21
Figura 3.8	Interface central do software Somachine V4.1.....	22
Figura 3.9	Interface <i>Logic builder</i>	23
Figura 3.10	Interface do Vijeo <i>designer</i>	24
Figura 3.11	Dimensões do visor HMISTU855.....	25
Figura 4.1	Capacidades de condução de correntes, em amperes, para os métodos de referência A1, A2, B1, B2, C e D.....	27
Figura 4.2	Esquema de fiação dos dispositivos.....	28
Figura 4.3	Organização final dos dispositivos dentro do painel.....	29
Figura 4.4	Arranjo dos dispositivos na tampa do painel.....	30
Figura 4.5	Interconexão dos dispositivos.....	30
Figura 4.6	Parafuso para aterramento do servo-drive.....	31
Figura 4.7	Visão geral da alimentação para o dispositivo.....	32
Figura 4.8	Diagrama de cabeamento dos servo-drives.....	32
Figura 4.9	Módulo CN2 do servo-drive.....	33
Figura 4.10	Luzes indicadoras do controlador de movimentos.....	34
Figura 4.11	Diagrama de alimentação do controlador de movimentos.....	35
Figura 4.12	Ilustração do plug do HMI.....	36
Figura 4.13	Conexão do plug ao HMI.....	36
Figura 4.14	Modelo para as entradas de alimentação do servo-motor.....	37

Figura 4.15 Pinos de conexão do cabo de alimentação do servo-motor.....	37
Figura 4.16 Configuração do cabo de alimentação do servo-motor.....	38
Figura 4.17 Composição do cabo de conexão com os terminais CN10 e CN11.....	39
Figura 4.18 Conexão do cabo de alimentação ao terminal do servo-drive.....	40
Figura 4.19 Conexão para o terminal <i> Holding brake </i>	41
Figura 4.20 Vista da conexão do cabo de energização dos motores aos módulos CN10 e CN11.....	41
Figura 4.21 Módulo CN6.....	42
Figura 5.1 Módulo CN3.....	43
Figura 5.2 Porta CAN.....	44
Figura 5.3 Configuração da taxa de transferência de dados.....	45
Figura 5.4 Configuração do endereço do nó do servo-drive.....	45
Figura 5.5 Configuração do mapeamento PDO.....	46
Figura 5.6 Configuração do escalonamento do motor.....	46
Figura 5.7 Configuração do período de ciclo para transferência de dados.....	47
Figura 5.8 Interface do programa utilizado para o teste de movimentação dos eixos.....	48
Figura 5.9 Fluxograma do algoritmo.....	49
Figura 5.10 Bloco de função MC_Reset.....	51
Figura 5.11 Bloco de função MC_Power.....	51
Figura 5.12 - Bloco de função R_Trig.....	52
Figura 5.13 - Bloco de função MC_Home.....	52
Figura 5.14 Bloco de função BLINK.....	53
Figura 5.15 Bloco de função MC_MoveAbsolute.....	54
Figura 5.16 Bloco de função MC_ReadActualPosition.....	55
Figura 6.1 Interface do usuário.....	57
Figura 6.2 Mensagem de erro após a realização do bloco de função MC_Home. Erro para o eixo X.....	57
Figura 6.3 Representação da perda de conexão.....	58
Figura 6.4 Mensagem de erro da aba Log.....	59
Figura 6.5 Primeiro teste de posicionamento.....	59
Figura 6.6 Resultado da multiplicação pelo fator de correção.....	60
Figura 6.7 Resultado do eixo X para o primeiro teste de movimentação.....	60
Figura 6.8 Resultado do eixo Y para o primeiro teste de movimentação.....	61
Figura 6.9 Resultado do eixo Z para o primeiro teste de movimentação.....	61
Figura 6.10 Segundo teste de posicionamento.....	61
Figura 6.11 Resultado do eixo X para o segundo teste de movimentação.....	62
Figura 6.12 Resultado do eixo Y para o segundo teste de movimentação.....	62
Figura 6.13 Resultado do eixo Z para o segundo teste de movimentação.....	63
Anexo I.1 Desenho técnico do MAXR23-S42-H42-C42.....	68

Anexo II.1 Característica dos atuadores	69
Anexo III.1 Configuração dos servo-motores.....	70
Apêndice I Programa	71
Apêndice II.1 Tampa	83
Apêndice II.2 Caixa.....	84
Apêndice II.3 Arranjo dos dispositivos no painel.....	85
Apêndice III.1 Desenho técnico da instalação elétrica	86
Apêndice IV.1 Dispositivos instalados na tampa do painel de controle (vista frontal)	88
Apêndice IV.2 Dispositivos instalados na tampa do painel de controle (vista traseira)	89
Apêndice IV.3 Dispositivos instalados dentro do painel de controle.....	90
Apêndice IV.4 Vista geral do robô MAXR23-S42-H42-C42.....	91
Apêndice IV.5 Vista dos eixos Y e Z do robô	91
Apêndice IV.6 Vista do eixo X do robô.....	92

LISTA DE TABELAS

Tabela 3.1 Principais características do robô	15
Tabela 3.2 Característica dos servo-motores.....	18
Tabela 3.3 Visão geral das conexões do servo-drive	20
Tabela 3.4 Descrição dos componentes do controlador de movimentos.....	22
Tabela 4.1 Descrição das luzes indicadoras do controlador de movimentos	34
Tabela 4.2 Descrição dos pinos de entrada do servo-motor.....	37
Tabela 4.3 Especificações do cabo de alimentação dos servo-motores	39
Tabela 5.1 Especificações dos cabos de encoder	43
Tabela 5.2 Características do encoder do servo-motor	44
Tabela 5.3 Relação do tamanho do cabo CAN com a taxa de transferência de dados.....	45
Tabela 5.4 Tempo mínimo para transferência de dados para conexão CANopen.....	47
Apêndice III.2 Legenda dos dispositivos utilizados no desenho técnico da instalação elétrica	87

LISTA DE SÍMBOLOS

Símbolos latinos:

a	Distância	[m]
c	Cosseno	
$c_0, c_1, c_2, c_3, c_4, c_5$	Constantes	
d_i	Posição inicial	[m]
d_f	Posição final	[m]
s	Seno	
t_i	Tempo inicial	[s]
t_f	Tempo final	[s]

Símbolos gregos:

α	Ângulo	[graus]
θ	Ângulo	[graus]

Abreviações:

ABNT	Associação Brasileira de Normas Técnicas
CLP	Controlador Lógico Programável
CSP	<i>Cyclic Synchronous Position</i>
GRACO	Grupo de Automação e Controle
IEC	<i>International Electrotechnical Commission</i>
NBR 5410/2004	Norma Brasileira 5410 de 2004
PDO	<i>Process Data Object</i>
UnB	Universidade de Brasília

CAPÍTULO 1 – Introdução

1.1 Histórico e definições

O termo “robótica” e os assuntos que o cercam não possuem conceitos definidos. Em vários aspectos, pode-se conectar o estudo da ciência com áreas como inteligência artificial, automação e controle remoto. Para a engenharia, robótica é a ciência que estuda e desenvolve robôs. Mas o que ainda não existe é um conceito preciso de o que poderia vir a ser um robô [16].

Nos dias de hoje, o estudo da robótica se concentra grande parte em desenvolver a robótica industrial [16]. O alto custo em pesquisa e desenvolvimento de máquinas automatizadas faz o setor ser um dos poucos com recursos suficientes para patrocinar o desenvolvimento da ciência.

Diariamente, novas tecnologias desenvolvidas podem se assemelhar às condições básicas para que algo seja considerado um robô [7]. Um grande exemplo disso são os sistemas que existem em fábricas, que podem controlar todo o processo de manufatura. São as chamadas células de manufatura, em que um grupo de equipamentos e robôs são responsáveis pela produção em massa. O objetivo é ganhar aumento do controle sobre toda a produção fabril, fazendo com que o operador tenha controle sobre todas as máquinas que trabalham no chão de fábrica.

1.2 Motivação

Como dito anteriormente, o estudo e o desenvolvimento da robótica têm um custo muito elevado, o que não permite que seu progresso seja mais efetivo em áreas fora da indústria. Na indústria, os robôs utilizados são chamados de robôs manipuladores e tem funções diversas, como manuseio de materiais (transferência, carga e descarga); processos de fabricação (soldagem, pintura); e montagem.

No geral, a aplicação dos robôs nas indústrias se dá em áreas em que a utilização de mão humana é perigosa, ou em trabalhos repetitivos. O que promove maiores pesquisas em áreas que buscam o aperfeiçoamento da precisão dinâmica de braços robóticos, bem como eficiência energética e diminuição de custos com manutenção.

Recentemente, o Grupo de Automação e Controle da UnB (GRACO) adquiriu o manipulador cartesiano MAXR23-S42-H42-C42, da Schneider Electric. A oportunidade de fazer a montagem, instalação mecânica e elétrica e programação desse robô é de grande valor para mostrar os conhecimentos adquiridos durante o curso de Engenharia Mecatrônica. Esse projeto está voltado para a área de robótica industrial, e obter conhecimentos sobre essa área é de grande importância tanto para o desenvolvimento profissional quanto para o desenvolvimento de futuras pesquisas. Além disso, a execução do projeto irá auxiliar o laboratório a automatizar processos para torná-los mais rápidos e eficientes. Por exemplo, a adição de uma tocha de soldagem em seu efetuador terminal possibilitará sua aplicação no processo de soldagem de materiais, além de poder ser utilizado como ferramenta para estudos em matérias como robótica industrial e elementos de automação.

1.3 Objetivo

O objetivo deste projeto é fazer a instalação e programação de um robô que trabalha nos eixos X, Y e Z (robô cartesiano), assim como a instalação e integração dos dispositivos de controle desse robô. Também será feita a conexão do robô ao painel elétrico e de controle e à rede de energia do laboratório Graco. A ideia principal será tornar o manipulador útil para a automatização de processos em que o robô poderá ser utilizado.

1.4 Estrutura do trabalho

Tendo em vista que o foco deste trabalho é realizar a instalação do robô, sua programação e conexão com o painel de controle, é importante verificar a literatura da área para se apoiar nos conceitos e atingir a proposta do trabalho.

O capítulo dois apresentará os principais conceitos sobre robótica industrial. Também serão estudados os sensores, atuadores e elementos que compõem o painel de controle. Por fim, serão apresentados os conceitos sobre a interface CANopen e métodos de planejamento de trajetórias.

Na sequência, no capítulo três serão abordados os elementos escolhidos para fazer a conexão do painel elétrico e de controle à rede de energia e ao robô. Neste capítulo será também introduzido o software utilizado para programação do controlador de movimentos, o Somachine V4.1

O capítulo quatro apresentará como ocorreu a instalação elétrica dos dispositivos no painel de controle.

No capítulo cinco serão abordadas as fases de teste de locomoção dos eixos, bem como a configuração dos dispositivos tanto no software quanto nos servo-drives. O capítulo também apresentará o algoritmo realizado para fazer um programa de movimentação dos eixos.

O capítulo seis mostrará os resultados experimentais obtidos por meio da execução do programa desenvolvido no software específico para o controle do robô, além de mostrar os resultados dos testes de movimentação e da instalação dos dispositivos.

CAPÍTULO 2 – Fundamentação Teórica

2.1 Robótica Industrial

Os robôs industriais, usados em operações em que se deseja alcançar maior velocidade e demandam acurácia e trabalho repetitivo, possuem uma herança muito longa na história da indústria da manufatura, e operam em ambientes relativamente estáticos e em grande quantidade. [3]

Robôs industriais têm se estabelecido na indústria manufatureira por mais de 30 anos, e vêm sendo utilizados para trabalhos como empilhamento, seleção, pintura, ordenação, soldagem, entre outros [3]. A figura 2.1 mostra um exemplo de manufatura robotizada para produção de automóveis. Atualmente, há novos desenvolvimentos na área de robotização de indústrias, cujo trabalho é perigoso para o homem, podendo então serem substituídos por manutenção, inspeção e reparos feitos através do controle de robôs a distância. Além disso, enquanto a fábrica está em funcionamento, operadores podem reprogramar o robô para realizar sua próxima tarefa.

Os sistemas robóticos são integrados a outros sistemas de automação de uma indústria de modo que permitem produção *just-in-time* e suportam novos níveis de manufatura personalizada de forma economicamente viável. Uma vez programada, uma linha de produção robotizada pode criar diferentes modelos ou fabricar variantes dos mesmos produtos, como a demanda exigir.



Figura 2.1 - Manufatura robotizada para produção de automóveis [19]

Atualmente, existem sistemas robóticos disponíveis no mercado que possuem alto nível de acurácia, velocidade, repetibilidade e agilidade. Assim, um novo foco tem sido adicionado na interação humano-robô, por exemplo, a simplificação da programação de sistemas robóticos e redução dos perigos causados a trabalhadores.

Os avanços em *machine learning*, em particular, em visão computacional, tem permitido que, sistemas robotizados realizem o trabalho que antes só poderia ser feito por operadores humanos, como *pick and place*

e controle de qualidade. Essas tecnologias têm sido empregadas de modo a dar suporte para um melhor nível de automação adaptativa. [1]

2.1.1 Componentes de robôs

Um sistema robótico demanda elementos em sua composição, que, integrados, formam o robô como um todo [17]: manipulador ou explorador, efetuador terminal, atuadores, sensores, controlador, processador e software.

O manipulador é o corpo principal do robô e consiste nas ligações, das articulações e de outros elementos estruturais. Um exemplo de manipulador pode ser visto na figura 2.2. Porém, sem os outros elementos o manipulador não é um robô.

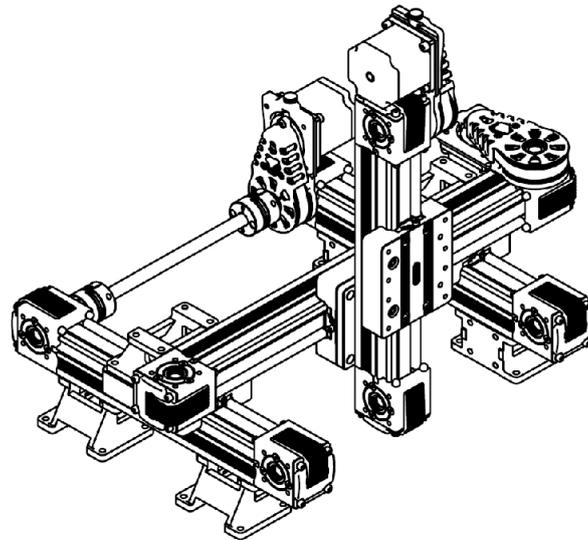


Figura 2.2 - Exemplo de um manipulador linear [20]

O efetuador terminal, visto na figura 2.3, é a parte que está ligada à última articulação de um manipulador que, em geral, movimenta objetos, faz conexões com outras máquinas ou executam o trabalho postulado. No geral, a mão de um robô é projetada para se conectar a atuadores finais especializados feitos especificamente para um propósito.

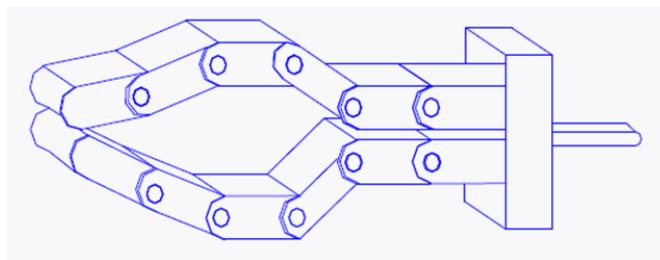


Figura 2.3 - Efetuador terminal modelo garra articulada [22]

A movimentação dos elos e das articulações do robô é feita pelos atuadores. Os tipos mais comuns de atuadores utilizados são os motores de passo, atuadores elétricos, atuadores pneumáticos e atuadores hidráulicos. Um exemplo de atuador elétrico pode ser visto na figura 2.4.



Figura 2.4 - Atuador linear [23]

Para que seja feita a coleta de informações sobre o estado interno do robô ou a comunicação com o ambiente externo são utilizados os sensores. Esses dispositivos enviam informações sobre as articulações ou elos para o controlador que determina a configuração do robô. Um exemplo de sensor, o encoder, pode ser visto na figura 2.5.

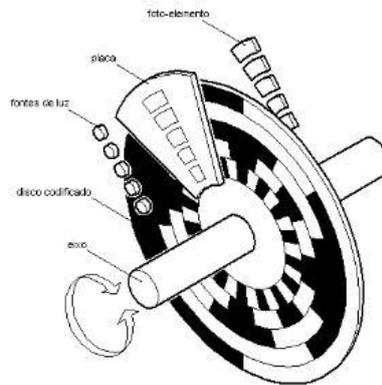


Figura 2.5 - Encoder absoluto e seu disco codificado [21]

O controlador, visto na figura 2.6, é responsável por fazer o controle dos movimentos do robô. Esse instrumento recebe os dados do computador, controla os movimentos dos atuadores e coordena os movimentos com a informação de realimentação dada pelos sensores.



Figura 2.6 - Controlador de movimentos multi-eixos [12]

O cálculo dos movimentos das articulações do robô é feito por um processador. Ele determina o quanto e quão rápido cada articulação deve se mover para atingir o local e a velocidade desejados, além de supervisionar as ações coordenadas do controlador e dos sensores. O processador é geralmente um computador que pode estar integrado ao mesmo sistema do controlador, e requer um sistema operacional, programas e equipamentos periféricos.

Há três grupos de programas de software que são usados em um robô [17]. O primeiro é o sistema operacional do processador; além desse, existe o software robótico, que é responsável por calcular os movimentos necessários de cada articulação com base na cinemática do robô; e, por último, tem-se o conjunto de rotinas orientadas à aplicações e programas desenvolvidos para uso do robô ou de seus periféricos para tarefas específicas, como montagem, carregamento de máquinas e manuseio de materiais.

Robôs possuem uma variedade de formas complexas: braços robóticos de formas variadas, veículos que possuem vários possíveis arranjos de rodas ou pernas ou dispositivos que, embora sejam robóticos, não são veículos ou braços [2]. Um exemplo pode ser visto na figura 2.7. De fato, em se tratando de aspectos gerais sobre a geometria de um robô ou do seu design, um robô industrial é uma máquina que tem por objetivo inicial mover objetos. O que ele move pode ser uma peça, uma ferramenta ou uma mercadoria. No geral, uma carga, e essa carga é, normalmente, um objeto com forma definida e rígida.

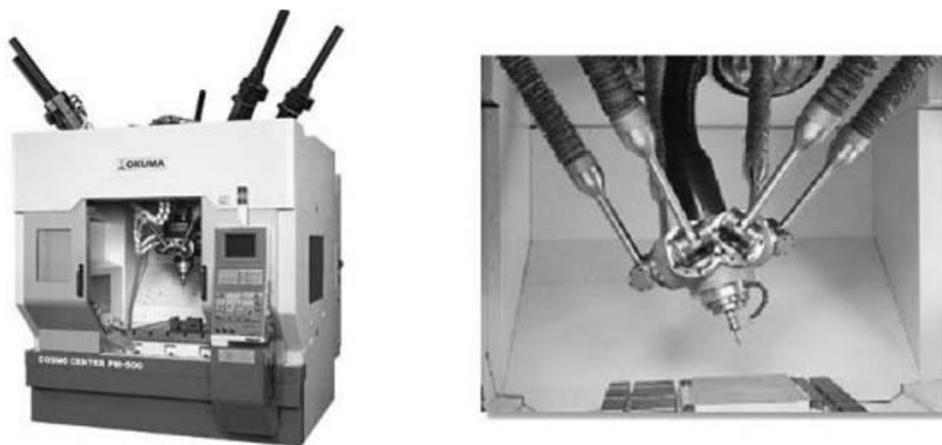


Figura 2.7 - Robô paralelo com 5 graus de liberdade [24]

Se o propósito do robô é fazer a movimentação de uma ferramenta em relação a um ponto fixo, são necessárias seis dimensões para realizar a tarefa [17]. Dessas dimensões, três são responsáveis por localizar um ponto no espaço, as outras três são responsáveis por mostrar a orientação do robô. Essa definição será útil quando precisarmos calcular a cinemática do robô utilizado no projeto.

Assim como é necessário haver seis dimensões para que seja possível localizar exatamente a posição de um robô no espaço e sua orientação, é necessário que o robô possua seis graus de liberdade afim de que o robô possa posicionar um objeto livremente e orientá-lo dentro de sua área de trabalho. Os graus de liberdade de um robô referem-se aos movimentos que o robô pode realizar nos eixos X, Y e Z. Caso o robô possua menos que seis graus de liberdade, ele só poderá realizar movimentos e ir a posições que suas articulações deficientes permitirem. O efetuator terminal não é considerado na hora de se definir os graus de liberdade de um robô.

Dependendo do tipo de utilização que terá o robô, podem ser escolhidos diferentes tipos e tamanhos de articulações. De acordo com essas características, é definido o espaço de trabalho do robô, que se refere aos pontos que podem ser atingidos pelo efetuador terminal. O espaço de trabalho de um robô pode ser encontrado através de equações que definem suas articulações e ligações, e depende das suas limitações (como alcance dos movimentos de cada articulação) ou empiricamente, através da movimentação virtual de cada articulação e combinando todo o espaço que ele pode alcançar, subtraindo o que ele não pode alcançar. É de extrema importância saber o espaço de trabalho de um robô antes de sua utilização, já que é necessário ter conhecimentos antecipados sobre as características do robô para saber se ele se encaixa nos requisitos de trabalhos desejados.

2.2 Tecnologias de controle, sensores, atuadores e drives de comando

Como foi visto na seção 2.1, um robô deve possuir os elementos de controle, sensores e atuadores. Caso contrário será um simples manipulador.

2.2.1 Atuadores

Os atuadores são como os músculos, e movem ou giram os elos para alterar a configuração do robô. Esse elemento deve ter potência suficiente para acelerar e desacelerar os elos e para transportar cargas, além disso deve ser leve, econômico, ágil e de fácil manutenção [17]. Os tipos mais comuns de atuadores são:

- Motores elétricos;
- Atuadores hidráulicos;
- Atuadores pneumáticos.

Na robótica, os tipos de atuadores mais utilizados são os motores elétricos, em especial os servo-motores [17].

Quando o projeto de um sistema robótico é feito, as características nominais dos atuadores devem ser levadas em consideração, como peso, potência, relação peso-potência, pressão e tensão de operação e temperatura [18]. Como os atuadores são, na maioria das vezes, colocados diretamente nas articulações, e movem-se junto com elas, o peso do atuador age como uma carga sobre os atuadores anteriores e deve ser acelerado e desacelerado por eles. Por conta disso, requisitos de energia e atuadores com um peso determinado devem ser considerados.

2.2.2 Sensores

Os sensores são dispositivos utilizados para controle de realimentação interno, bem como para interação externa com o ambiente [17]. Quando os elos e articulações se movem, sensores enviam sinais para o controlador, permitindo que ele determine os valores articulares.

Quando o projeto de um sistema robótico é feito, as características dos sensores devem ser levadas em consideração, como:

- Tamanho: É importante assegurar que haja espaço suficiente para alocação dos sensores, já que estes se moverão junto com as articulações;
- Tipo de saída: pode ser digital ou analógica;
- Resolução: Tamanho do passo mínimo dentro do intervalo de medição do sensor;
- Sensibilidade: Proporção de uma variação na saída em resposta a uma variação na entrada;
- Gama: Diferença entre a maior e menor saídas que um sensor pode reproduzir;
- Tempo de resposta: Tempo que a saída de um sensor requer para chegar a uma determinada porcentagem da variação total, normalmente 95%.

Além desses requisitos, é importante também que requisitos como custo, peso, interfaceamento, linearidade, confiabilidade, precisão e repetibilidade sejam verificados.

Os principais sensores usados na área da robótica e fabricação automática são [17]:

- Sensores de posição;
- Sensores de velocidade;
- Sensores de aceleração;
- Sensores de força e pressão;
- Sensores de torque;
- Sensores de luz e infravermelho;
- Sensores de toque;
- Sensores de proximidade;

2.2.3 Tecnologias de controle

Um sistema de controle é utilizado para alterar o comportamento de um dispositivo, máquina ou processo [18]. No geral, é o sistema que controla uma planta. A entrada do sistema é o comando enviado pelo controlador e a saída será o movimento do braço robótico. Para executar sua função, o sistema de controle utiliza sensores, drives, controlador de movimentos e interface homem-máquina.

Servo drive é um dispositivo eletrônico que consiste de quatro dispositivos principais: um motor; um driver; um controlador e um sistema de realimentação, mais comumente um *encoder* [30]. O driver e o controlador trabalham em conjunto de modo a determinar o que o motor deve fazer (controlador) e envia a energia necessária para que esse motor funcione (driver).

O controlador é responsável por calcular a trajetória requerida e mandar sinais de comando de baixa tensão ao driver, este por sua vez manda a tensão e corrente necessárias ao motor para que este realize o movimento desejado. O driver pode controlar torque, velocidade e posição, porém o servo-drive é mais comumente usado para controlar o torque.

Controladores de movimento são considerados o cérebro de um sistema de controle. Eles recebem a informação do programador, comparam com a informação recebida do sistema a ser controlado e faz os ajustes necessários para que a entrada e a saída do sistema operem em sincronia com erro mínimo.

Esses controladores também criam a trajetória em que o motor irá seguir de acordo com o comando enviado, bem como a velocidade em que esse processo irá ocorrer.

A interface homem-máquina (HMI – *Human-Machine Interface*) é uma interface que conecta o operador ao controlador em um sistema industrial, e inclui componentes eletrônicos para sinalização e controle de sistemas de automação. Algumas HMI's também fazem a tradução de dados em um sistema de controle industrial de modo que as informações do sistema possam ser lidas por um humano.

Através desse dispositivo, é possível que o operador veja o que está acontecendo em um chão de fábrica, por exemplo, através de desenhos esquemáticos e tomar decisões como aumentar/diminuir temperaturas, ligar/desligar dispositivos.

2.3 Protocolo de comunicação CANopen

Muitos dos sistemas *fieldbus* e protocolos de controle por rede estão sendo desenvolvidos com o intuito de se satisfazer o requisito de trabalho em tempo real [3]. Atualmente existem alguns protocolos de rede padronizados pelo ISO que têm o objetivo de preencher essa característica. Um desses protocolos é o CAN (*Controller Area Network*), que provê uma solução geral para controle de redes para muitos requisitos industriais, e pode ter sua aplicação distribuída em um equipamento ou em uma célula de trabalho [4].

Embora o protocolo CAN tenha sido desenvolvido para trabalhar na indústria automotiva, ele vem sendo largamente empregado em outros tipos de indústrias devido aos seus recursos, como rápida transmissão e consistência de dados, mecanismos de detecção de erros, robustez e a relação preço/desempenho é satisfatória. E alguns protocolos que trabalham em camadas vem sendo desenvolvidos para trabalhar especificamente com o CAN, como o CANopen [5].

O protocolo CAN é um protocolo de comunicação serial, definido pelo ISO 11898, que possui arquitetura multi-mestre baseada em broadcast (transmissão de dados a distância) e abrange as camadas física e de enlace. A arquitetura CAN padrão pode ser vista na figura 2.8. A transmissão de dados pode chegar a até 1 Mbits/s, além de poder ser implementado em sistemas de tempo real e possuir mecanismos robustos de detecção de erros. A substituição de componentes da rede pode ser feita durante a operação do sistema [6].

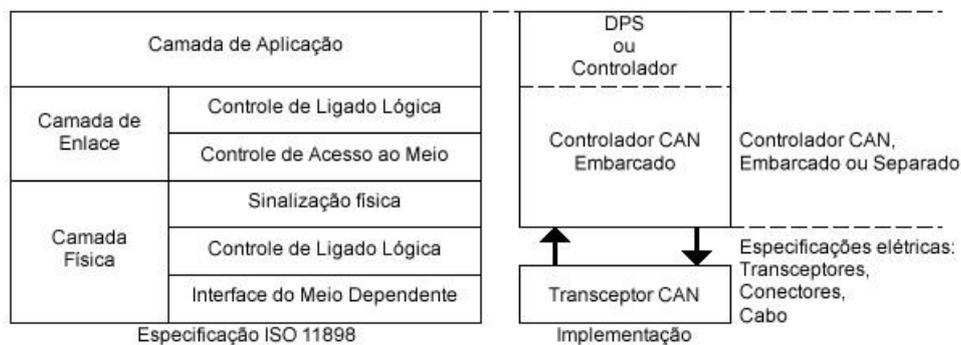


Figura 2.8 - Arquitetura CAN padrão [6]

O protocolo utiliza o CSMA (*Carrier Sense Multiple Access*) com detecção de colisão e arbitragem na prioridade da mensagem, que são resolvidas pela arbitragem de lógica binária, com base na prioridade pré-programada de cada mensagem [6]. Seus dispositivos possuem identificadores de 11 bits ou 29 bits, também chamados de CAN 2.0A, visto na figura 2.9, definido no ISO 11898, e CAN 2.0B, visto na figura 2.10, definido no ISO 11989, respectivamente.

Resistores de *pull-up* são utilizados para manter o estado recessivo e são conectados aos dois fios do meio de transmissão da rede CAN, e nesse meio de transmissão é avaliada a diferença de tensão entre os fios. Como são utilizados esses resistores de *pull-up*, a rede recebe um número constante de bits recessivos sem que seja transmitida nenhuma informação. Assim, para marcar o começo de uma transmissão é necessário um estado lógico diferente [5].

O acesso ao barramento é um evento dirigido e ocorre de maneira aleatória, porém são atribuídas prioridades às mensagens. Assim, se dois nós tentarem acessar o barramento ao mesmo tempo, o evento com maior prioridade acessará o barramento e transmitirá a mensagem, sem que o evento não prioritário seja destruído, já que o acesso é implementado através de arbitragem lógica não destrutiva [6].

Os campos existentes nos pacotes de dados do protocolo CAN 2.0A são:

- SOF: Início de quadro e possui um único bit dominante, que marca o início da mensagem e é utilizado para sincronizar no barramento após ficar ocioso;
- Identificador: Valor que estabelece a prioridade da mensagem, a mensagem com menor valor possui maior prioridade;
- RTR: Único bit dominante para pedido de transmissão remota, possui nível lógico alto. Todos os nós recebem o pedido, mas o nó específico é determinado pelo Identificador. A resposta chega a todos os nós;
- IDE: Único bit dominante que significa que o campo identificador não segue o padrão estendido;
- r0: Bit reservado;
- DLC: Dado de 4 bits que contêm o número em bytes do dado que está sendo transmitido;
- Dado: Dado da aplicação a ser transmitido. Pode chegar até 64 bits;
- CRC: teste de redundância cíclica de 16 bits, que faz a detecção de erros;
- ACK: todo nó que receber uma mensagem livre de erros substitui o bit recessivo desse campo por um bit dominante. Em caso de erro, o bit é deixado como está e a mensagem é descartada. O nó transmissor repete a mensagem após a arbitragem das prioridades. Esse campo possui dois bits, um para escrita e outro como delimitador;
- EOF: campo com sete bits que indicam o final de uma mensagem e verificam o erro de bit *stuffing*;
- IFS: Campo de sete bits que indica o tempo necessário para enviar a mensagem para a aplicação (memória).

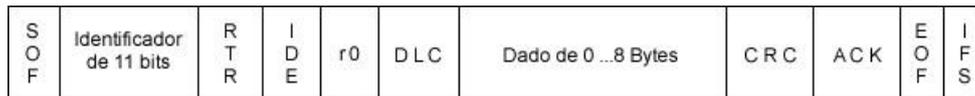


Figura 2.9 - CAN 2.0A [6]

Existem 3 campos adicionais para o CAN 2.0B, que são:

- SSR: Bit único que substitui o RTR na posição da mensagem como uma área reservada no formato estendido;
- IDE: Bit recessivo que indica a extensão do identificador com acréscimo de 18 bits;
- r1: Bit reservado adicional

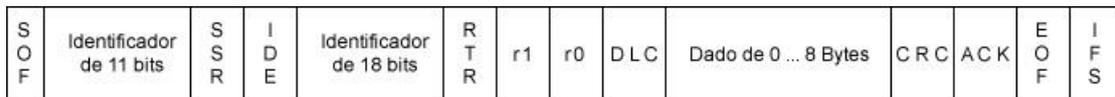


Figura 2.10 - CAN 2.0B [6]

As mensagens transmitidas podem ser de quatro tipos:

1. Dado: tipo mais comum, e inclui os campos de arbitragem, dado, RCR e ACK;
2. Remota: tem o propósito de solicitar dado de outro nó da rede. É similar à mensagem de dado, porém o campo RTR contém um bit recessivo e o campo de dados é vazio;
3. Erro: Tipo especial que viola o formato de mensagem CAN e é transmitida caso seja detectado um erro em uma mensagem por algum nó. Uma mensagem é considerada sem erro quando o último bit no campo EOF é recebido como bit recessivo livre de erro, se o bit no EOF for dominante, isso faz o transmissor reenviar a mensagem;
4. Sobrecarga: Transmitido quando um nó está ocupado e é usado, principalmente, para fornecer atraso extra entre mensagens. Tem formato similar à mensagem de erro.

A checagem de erro é feita por cinco métodos diferentes, três em nível de mensagem e dois em nível de bit. Caso uma mensagem falhe em um desses métodos, ela não é aceita e uma mensagem de erro é gerada por qualquer nó que perceba a mensagem errada, já que o CAN trabalha em broadcast, que faz o transmissor reenviar a mensagem até que ela seja aceita.

CANopen é um protocolo de comunicação de alto nível que padroniza a comunicação entre dispositivos e aplicações da rede. Esse protocolo, baseado no CAN (como dito anteriormente), trata do endereçamento, roteamento, confiabilidade fim-a-fim, sincronização, padronização e representação dos dados [6]. A camada de aplicação é responsável pela descrição de como configurar, transferir e sincronizar os dispositivos da rede.

A rede CANopen deve possuir um mestre, que é responsável por gerenciar a rede. Uma rede pode possuir até 127 escravos (nós). Todo nó da rede possui uma lista de objetos denominada de dicionário de objetos, que contém objetos de comunicação (COB) responsáveis pela comunicação entre dispositivos da rede. Esses objetos são:

- Objeto de serviço de dados, responsável pelo acesso direto ao dicionário de objetos de um dispositivo da rede;
- Objeto de processamento de dados, usado para acessar os dados de um dispositivo;
- Objeto de emergência, responsável pelo envio de mensagens para indicar a ocorrência de erros no dispositivo;
- Objeto de sincronização, que permite a um dispositivo enviar uma mensagem de sincronização para toda a rede, periodicamente;
- Objeto de gerenciamento de rede, para o mestre da rede gerenciar os seus serviços de controle do dispositivo e serviço de controle de erros nos nós da rede.

O protocolo CANopen utiliza a mensagem padrão do CAN com o campo Identificador dividido em duas partes:

1. Representada por quatro bits, é usada para a identificação da função;
2. Possui sete bits, é usada para identificação do nó.

A união dessas duas partes mais o campo RTR resulta no Identificador de Objetos de Comunicação (COB-ID).

2.4 Planejamento de trajetória

Planejamento de trajetória se refere à forma como um robô é movido no espaço, de um local para o outro de forma controlada. Esse planejamento pode ser feito tanto no espaço articular quanto no espaço cartesiano [17].

A descrição “*espaço-articular*” se refere à descrição do movimento a ser feito pelo robô por seus valores articulares. Essa descrição é feita com base na cinemática inversa do robô, que permite calcular os deslocamentos articulares totais que o robô precisa fazer para chegar até um novo local. Os valores articulares calculados podem ser usados pelo controlador para conduzir as articulações do robô para seus novos valores, ou seja, mover os atuadores para sua nova posição [17].

A trajetória de espaços cartesianos é mais facilmente visualizável, já que é o espaço de operação humano [17]. Assim, é mais fácil enxergar qual deve ser a trajetória do efetuador terminal do robô. Entretanto, apesar dessa facilidade na visualização, trajetórias no espaço cartesiano são mais dispendiosas computacionalmente e demandam tempo de processamento mais rápido para uma resolução semelhante à de trajetórias articulares. Por exemplo, sem o devido cuidado, é possível que seja especificado uma trajetória para o robô em que ele mova para si mesmo ou saia do seu espaço de trabalho.

Para auxiliar no desenvolvimento do planejamento de trajetórias para robôs, existem alguns esquemas diferentes tanto para o espaço articular quanto para o espaço cartesiano [17]. Neste projeto, será trabalhado com o espaço articular. Para essa configuração, existem dois tipos principais de esquema de cálculo:

- Planejamento de trajetórias por polinômios de terceira ordem;

- Planejamento de trajetórias por polinômios de quinta ordem;

2.4.1 Planejamento de trajetórias por polinômio de terceira ordem

Para este tipo de aplicação, a localização e a orientação iniciais do robô já estão definidas. Assim, utilizando as equações de cinemática inversa é possível definir a localização e ângulos articulares finais desejados [17]. Os movimentos de cada articulação do robô são planejados separadamente. Assim, para cada articulação, tem-se:

- d_i : posição inicial;
- d_f : posição final;
- t_i : tempo inicial;
- t_f : tempo final;

Os quatro itens descritos permitem que seja calculado um polinômio de terceira ordem na forma:

$$\mathbf{d}(t) = \mathbf{c}_0 + \mathbf{c}_1 t + \mathbf{c}_2 t^2 + \mathbf{c}_3 t^3 \quad \text{Equação 2.1}$$

As condições iniciais e finais são:

$$d(t_i) = d_i;$$

$$d(t_f) = d_f;$$

$$d'(t_i) = 0;$$

$$d'(t_f) = 0;$$

sendo:

$$\mathbf{d}'(t) = \mathbf{c}_1 + 2\mathbf{c}_2 t + 3\mathbf{c}_3 t^2 \quad \text{Equação 2.2}$$

Com base nas condições iniciais e finais mencionadas, as equações podem ser escritas substituindo-se os valores iniciais e finais:

$$\mathbf{d}(t_i) = \mathbf{d}_i = \mathbf{c}_0 \quad \text{Equação 2.3}$$

$$\mathbf{d}(t_f) = \mathbf{c}_0 + \mathbf{c}_1 t_f + \mathbf{c}_2 t_f^2 + \mathbf{c}_3 t_f^3 \quad \text{Equação 2.4}$$

$$\mathbf{d}'(t_i) = \mathbf{c}_1 = \mathbf{0} \quad \text{Equação 2.5}$$

$$\mathbf{d}'(t_f) = \mathbf{c}_1 + 2\mathbf{c}_2 t_f + 3\mathbf{c}_3 t_f^2 = \mathbf{0} \quad \text{Equação 2.6}$$

A resolução simultânea das quatro equações permite obter os valores necessários para as constantes, o que permite calcular a posição articular em qualquer intervalo de tempo, que pode ser utilizada pelo controlador para conduzir a articulação para a posição determinada. Esse processo é realizado para cada articulação, mas a movimentação delas é feita de forma simultânea. No caso de serem especificados mais de dois pontos para que o robô percorra, as velocidades e posições finais na conclusão de cada segmento podem ser utilizados como condições iniciais para a resolução do próximo segmento. Entretanto, apesar de as velocidades e posições serem contínuas, acelerações não são, o que pode desenvolver algum problema. Nesse caso, é utilizado o planejamento de trajetória com polinômio de quinta ordem.

2.4.2 Planejamento de trajetórias por polinômio de quinta ordem

A utilização de polinômio de quinta ordem para o planejamento de trajetórias permite que, além da posição e da velocidade, a aceleração também seja considerada para a realização da movimentação do robô. Assim, a forma do polinômio é dada por:

$$d(t) = c_0 + c_1t + c_2t^2 + c_3t^3 + c_4t^4 + c_5t^5 \quad \text{Equação 2.7}$$

Derivando a equação, encontra-se a equação da velocidade dada por:

$$d'(t) = c_1 + 2c_2t + 3c_3t^2 + 4c_4t^3 + 5c_5t^4 \quad \text{Equação 2.8}$$

Por fim, derivando novamente encontra-se a equação para a aceleração:

$$d''(t) = 2c_2 + 6c_3t + 12c_4t^2 + 20c_5t^3 \quad \text{Equação 2.9}$$

Assim, utilizando as condições de contorno de posição, velocidade e aceleração iniciais e finais, é possível calcular o valor das constantes necessárias para o controle do robô.

CAPÍTULO 3 – Especificação Técnica dos Dispositivos

Este capítulo apresenta os dispositivos utilizados para fazer a correta alimentação e configuração do robô adquirido pelo GRACO.

3.1 Robô

A realização deste projeto deu continuidade ao trabalho realizado por Fernandes (2016), cujo objetivo foi desenvolver o projeto de instalação de um robô cartesiano. Seu projeto envolveu os aspectos de concepção e especificação dos dispositivos a serem adquiridos para a montagem mecânica e elétrica do robô. Em seu projeto de graduação foi escolhido o robô MAXR23-S42-H42-C42 da Schneider Electric, que pode ser visto na figura 3.1, bem como os dispositivos de controle e alimentação dos eixos, que serão apresentados nos capítulos adiante.

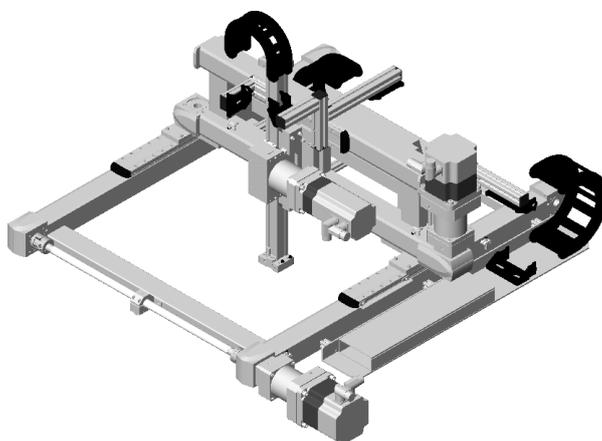


Figura 3.1 - Robô MAXR23-S42-H42-C42 da Schneider Electric [26]

A tabela 3.1 explicita as principais características do manipulador. O desenho técnico do robô pode ser visto de maneira detalhada no Anexo I.

Parâmetro	Unidade	Característica
Transferência de movimento	-	Correia dentada 25HTD-5M
Tipo de guia	-	Guia por rolamento
Carga típica	kg	15
Constante de alimentação	mm/rev	155
Diâmetro efetivo da polia da correia dentada	mm	49.338
Aceleração máxima	m/s ²	20
Percurso mínimo	mm	9
Repetibilidade	mm	+/- 0.1

Tabela 3.1 – Principais características do robô

Este robô é dotado de três bases que completam os eixos X, Y e Z. O eixo X é formado pela base MAXS2BB. As principais características dessa base são:

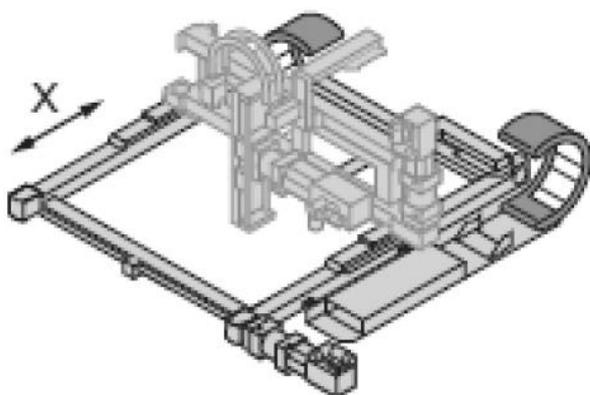


Figura 3.2 – Eixo X [26]

- Carga contínua: 75 kg;
- Força de operação: 1200 N;
- Velocidade máxima: 5 m/s;
- Aceleração: 20 m/s²;
- Torque: 280 N.m;
- Acurácia: +/- 0.1 mm;
- Servo-motor: BMH0702P06A2A

O eixo Y é formado pela base MAXH2BB, que possui as seguintes características:

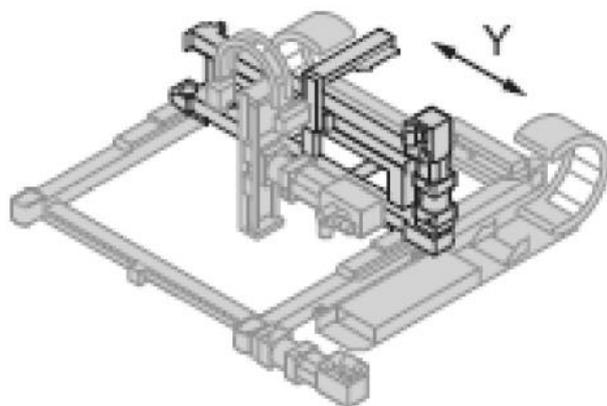
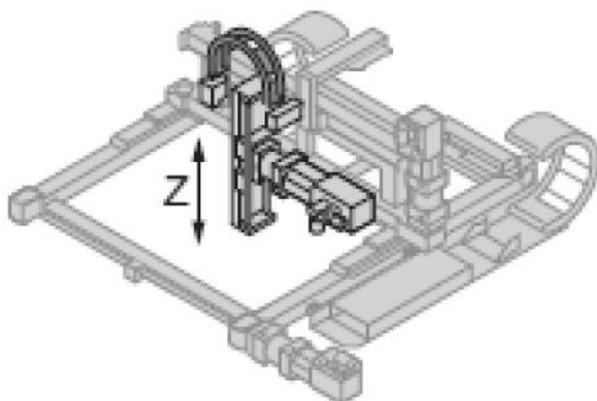


Figura 3.3 - Eixo Y [26]

- Carga contínua: 60 kg;
- Força de operação: 4215 N;
- Velocidade máxima: 5 m/s;
- Aceleração: 20 m/s²;
- Torque: 388 N.m;
- Acurácia: +/- 0.1 mm;
- Servo-motor: BMH0701P06A2A

Por último, o eixo Z é formado pela base CAS42BB, cujas características são:



- Carga contínua: 15 kg;
- Força de operação: 3540 N;
- Velocidade máxima: 5 m/s;
- Aceleração: 20 m/s²;
- Torque: 250 N.m;
- Acurácia: +/- 0.05 mm;
- Servo-motor: BMH0701P06F2A

Figura 3.4 - Eixo Z [26]

O volume de trabalho desse robô é de 800x800x500 mm, ou seja, o robô pode fazer movimentação livre dentro deste volume de atuação. O Anexo II mostra de maneira completa os detalhes de cada um dos atuadores do robô.

Como pôde ser visto no detalhamento dos atuadores do robô, cada eixo é movimentado por um servo-motor diferente. A nomenclatura de cada um desses servo-motores mostra que existem diferenças entre suas características e isso deve ocorrer porque cada eixo possui uma função específica para a movimentação do robô. Por exemplo, os servo-motores dos eixos X e Y se diferenciam, principalmente, em termos de torque. Como o eixo X é responsável por carregar toda a carga a ser movida pelo robô, além de carregar os eixos Y e Z, o torque fornecido por ele deve ser maior. Entre os eixos Y e Z, a principal diferença está no sistema de frenagem do eixo Z, para evitar que o eixo desça repentinamente ao ser desligado ou quando estiver movimentando uma carga. As principais características dos servo-motores podem ser vistas na tabela 3.2.

Servo-Motor	BMH0702P06A2A	BMH0701P06A2A	BMH0701P06F2A
Tamanho do flange	70 mm		
Comprimento	2 <i>stacks</i>	1 <i>stack</i>	
Enrolamento do motor	Otimizado em termos de torque e velocidade de rotação		
Eixo	Liso		
<i>Gearbox</i>	PLE 80	PLE60	
Redução	8:1		
Grau de proteção eixo/involúcro	IP54/IP65 (IEC60034-5)		
Sistema de Encoder	SEK37 – Giro único absoluto; 16 períodos de sen/cos por revolução		

Frenagem	Não		Sim
Versão da conexão	Conector angular de 90°, pode ser rotacionado		
Interface mecânica da montagem	<i>International IEC Standard</i>		
Torque contínuo	2,48 N.m	1,4 N.m	
Torque máximo	7,44 N.m	4,2 N.m	
Com tensão de alimentação de 230 Vac			
Velocidade nominal de rotação	3000 rpm		
Torque nominal	2,23 N.m	1,34 N.m	
Corrente nominal	2,7 A	1,75 A	
Potência nominal	0,7 kW	0,42 kW	

Tabela 3.2 - Característica dos servo-motores

No Anexo III pode ser visto com mais clareza o significado da nomenclatura dos servo-motores.

O manual dos servo-motores também explica que eles são alimentados por tensão AC, porém não podem ser conectados diretamente à rede geral de energia. Portanto, para sua correta alimentação foram escolhidos servo-drives, que são interfaces de potência e são responsáveis por gerir os sinais de entrada e saída.

3.2 Drives de acionamento

Para o projeto do sistema de controle do robô escolhido, foram utilizados 3 drives de acionamento. Os drives escolhidos são utilizados para fazer a alimentação e controle dos três eixos do robô.

Os servo-drives utilizados para realização deste trabalho são da família LXM da Schneider Electric.

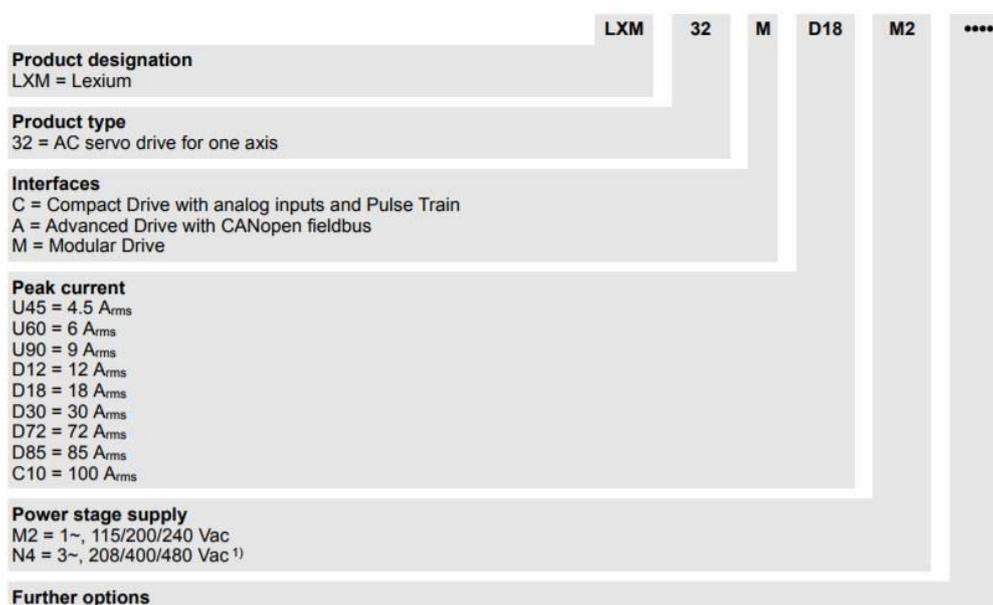


Figura 3.5 - Família LXM da Schneider Electric [14]

Os drives da família Lexium 32 consiste em vários modelos que realizam tarefas em diferentes áreas de aplicação. De acordo com o guia do usuário [14], este drive é a melhor opção para trabalhar com os servo-motores da família BMH, além de serem adequados para implementar soluções de alta performance para uma alta gama de requisitos de potência.

Os servo-drives escolhidos foram o LXM32MU90M2 e LXM32MD18M2. Pela figura 3.5 é possível ver as características gerais de cada um deles:

- LXM32MU90M2:

- a) Servo drive de controle de um eixo;
- b) Drive modular;
- c) Corrente de entrada de 4,5 ampères;
- d) Corrente de pico de saída de 9 ampères;
- e) Corrente nominal de 3A;
- f) Tensão de alimentação de 115/200/240 Vac;
- g) Potência nominal: 0,5 kW a 230 Vac;

- LXM32MD18M2:

- a) Servo drive de controle de um eixo;
- b) Drive modular;
- c) Corrente de entrada de 8,4 ampères;
- d) Corrente de pico de saída de 18 ampères;
- e) Corrente nominal de 6A;
- f) Tensão de alimentação de 115/200/240 Vac;
- g) Potência nominal: 1 kW a 230 Vac

Para o controle do robô foram escolhidos dois servo-drives LXM32MU90M2 para os eixos Y e Z. A potência fornecida por esse serve-drive é de 0,5 kW quando alimentado com uma tensão de 230 Vac, enquanto a potência nominal dos servo-motores dos eixos Y e Z é de 0,47 kW. Portanto, vê-se que esse servo-drive é suficiente para a alimentação do eixo. Para o eixo X foi escolhido o servo drive LXM32MD18MD, já que este eixo é o que carrega maior parte da carga e seu servo-motor tem potência nominal de 0,7 kW enquanto a potência fornecida por esse servo-drive é de 1 kW quando alimentado com uma tensão de 230 Vac. A figura 3.6 mostra a visão geral das conexões do dispositivo, enquanto a tabela 3.3 explica o que são essas entradas.

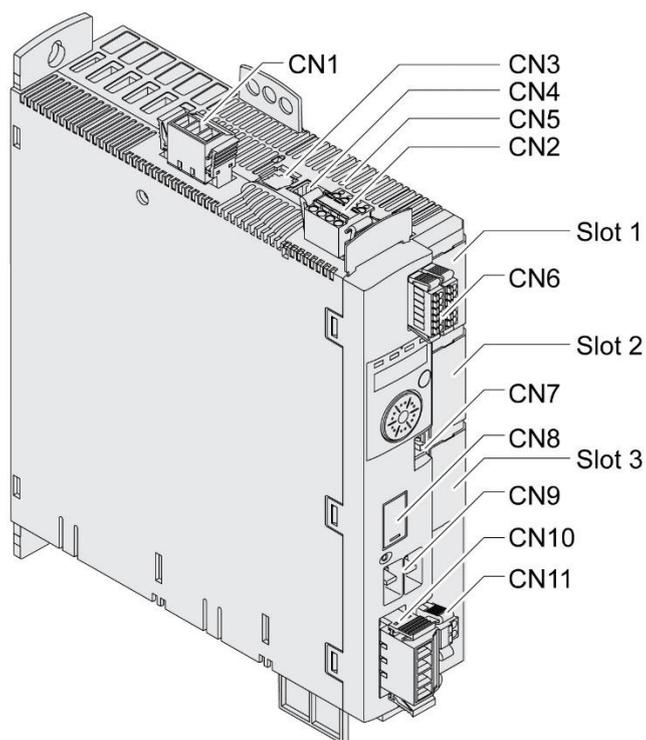


Figura 3.6 - Visão geral do dispositivo LXM32M [14]

Conexão	Atribuição
CN1	Fonte de alimentação (<i>Power stage supply</i>)
CN2	Fonte de controle e função de segurança STO (<i>Safe Torque Off</i>)
CN3	<i>Encoder 1 (Motor encoder)</i>
CN4	PTO (<i>Pulse Train Out – encoder simulation ESIM</i>)
CN5	PTI (<i>Pulse Train In – sinais A/B, P/D e CW/CCW</i>)
CN6	Entradas/Saídas digitais
CN7	Interface de comissionamento <i>Modbus</i>
CN8	<i>Braking resistor</i> externo
CN9	Conexão DC <i>bus</i> para operações em paralelo
CN10	<i>Motor phases</i>
CN11	<i>Holding brake</i> (Conexão com o freio do manipulador)
Slot 1	Módulo de segurança ou módulo E/S
Slot 2	<i>Encoder 2 (módulo encoder)</i>
Slot 3	Módulo <i>Fieldbus</i>

Tabela 3.3 - Visão geral das conexões do servo-drive

3.3 Controlador de movimentos

Um controlador de movimentos será necessário para o projeto afim de se fazer os ajustes nas informações enviadas aos servo-drives.

Para esse projeto, foi escolhido o controlador de movimentos Modicon LMC058LF42S0, da Schneider Electric. Este controlador de movimentos oferece uma solução otimizada para posicionamento de eixos devido ao software SoMachine V4.1, também da Schneider Electric, que inclui funções para automação de sistemas embarcados e possui uma interface ergonômica para configuração de eixos.

O controlador possui uma bateria de 3 volts e é alimentado por uma tensão de 24 volts. Também possui 42 portas de entrada/saída, como pode ser visto na figura 3.7 e na tabela 3.4.

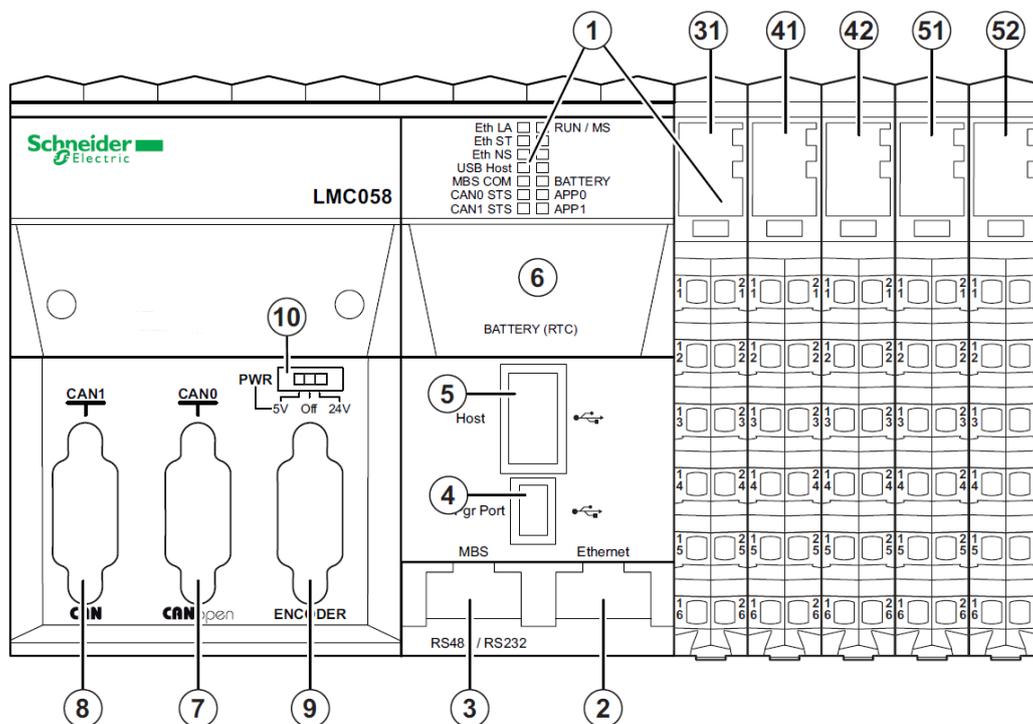


Figura 3.7 - Controlador de movimentos da família LMC058 [12]

Nº	Descrição
1	Luzes de status
2	Porta Ethernet
3	Serial Line
4	Porta USB
5	USB Host
6	Bateria
7	Porta CAN 0
8	Porta CAN 1
9	Interface de conexão com encoder
10	Switch do encoder
31	Módulo de controle de distribuição de energia

41 e 42	Módulos embarcados de entrada e saída/ 5 entradas rápidas, 2 entradas regulares 2 saídas rápidas
51	Módulos de entrada embarcados regular/ 12 entradas digitais
52	Módulos de saída embarcados regular/ 12 saídas digitais

Tabela 3.4 Descrição dos componentes do controlador de movimentos

Este dispositivo é, portanto, o controlador do robô. Utilizando o software Somachine V4.1, esse dispositivo é, então, programado e os comandos são enviados aos servo-drives. Os servo-drives são responsáveis por enviar os sinais de comando aos servo-motores dos eixos do robô, para realizar a movimentação dos atuadores.

3.4 Somachine V4.1

O software utilizado para a programação do controlador de movimentos LMC058 é o SoMachine V4.1.

Este software é uma solução profissional e eficiente com OEM's abertos que visam auxiliar no desenvolvimento, configuração e comissionamento de um processo em um único ambiente. Seus ambientes principais são o *Logic builder* e o *Vijeo designer*. A figura 3.8 mostra a interface central do software.

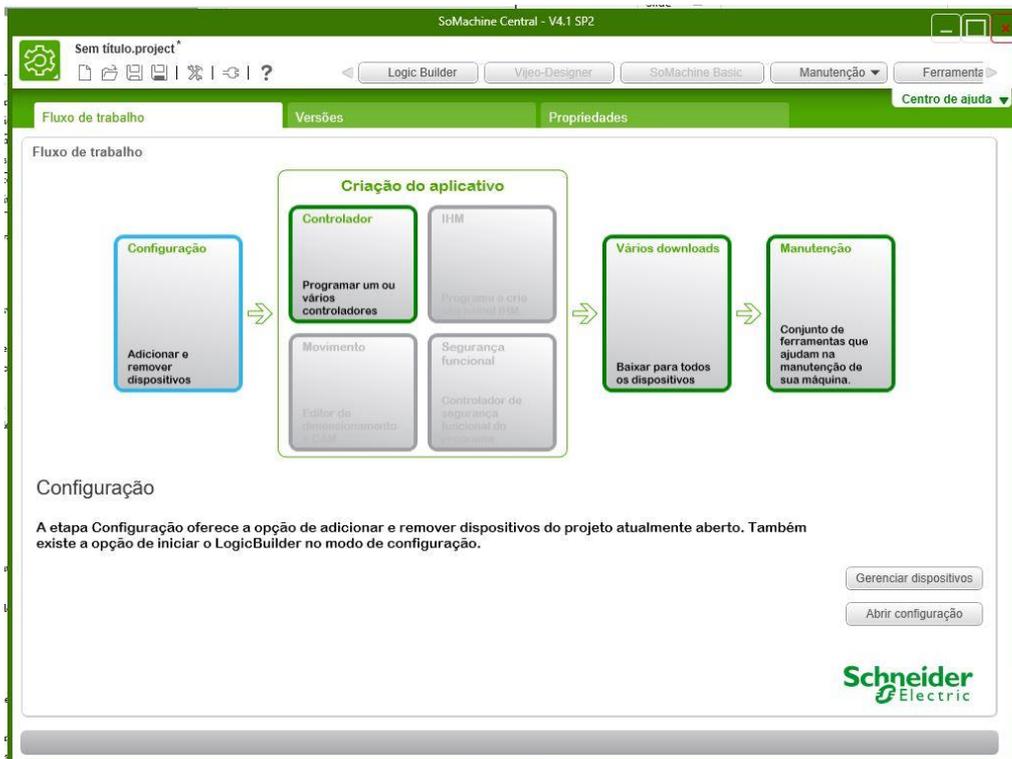


Figura 3.8 - Interface central do software Somachine V4.1

O *Logic builder* é o ambiente de programação da lógica e adição dos dispositivos utilizados no projeto. Esse ambiente tem suporte para todas as linguagens de programação de CLP's reconhecidas pelo IEC61131-3, sendo elas:

- IL: Lista de instruções;
- LD: Diagrama ladder;
- ST: Texto estruturado;
- FBD: Diagrama de blocos de função;
- SFC: Sequenciamento de funções gráficas

Além de ter suporte para linguagem CFC (Funções gráficas contínuas). O software também fornece suporte para os protocolos de comunicação:

- CANmotion Master dedicado à sincronização de dispositivos de movimento;
- CANopen Master;
- Ethernet;
- Serial line

O software também possui suporte para funções de entrada e saída, como Encoder Master e entradas/saídas embarcadas e suporte para até 21 tarefas.

A adição de dispositivos ao projeto é facilitada por meio do mecanismo *drag and drop* que o software possui. Ou seja, para adicionar novos dispositivos ao projeto, é necessário apenas que o usuário selecione o objeto a ser adicionado e sua programação já estará disponível para ser realizada. A interface *Logic builder* pode ser vista na figura 3.9.

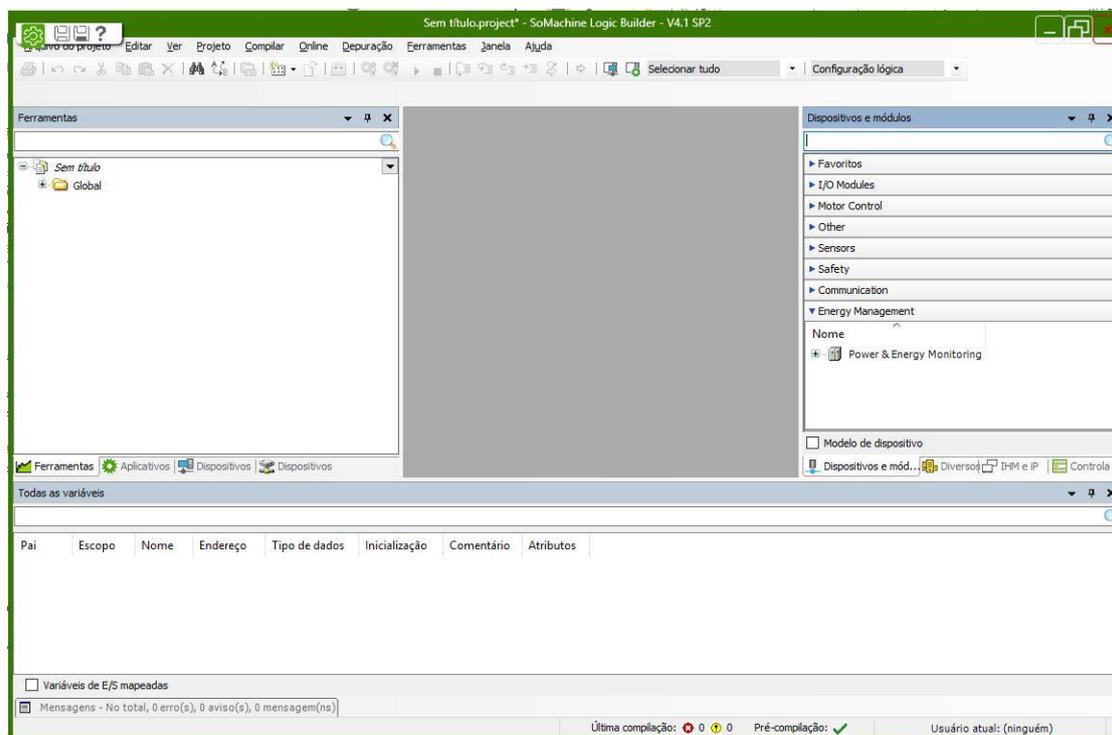


Figura 3.9 - Interface *Logic builder*

O *Vijeo designer* é o ambiente que permite a ligação do projeto ao HMI. Assim, o que for programado neste ambiente, poderá ser mostrado na tela da interface homem-máquina. Esse fato torna o controle do projeto mais fácil, já que permite que o programador seja avisado caso algum problema venha a ocorrer, além de permitir o controle do projeto por meio da tela. A interface pode ser vista na figura 3.10.

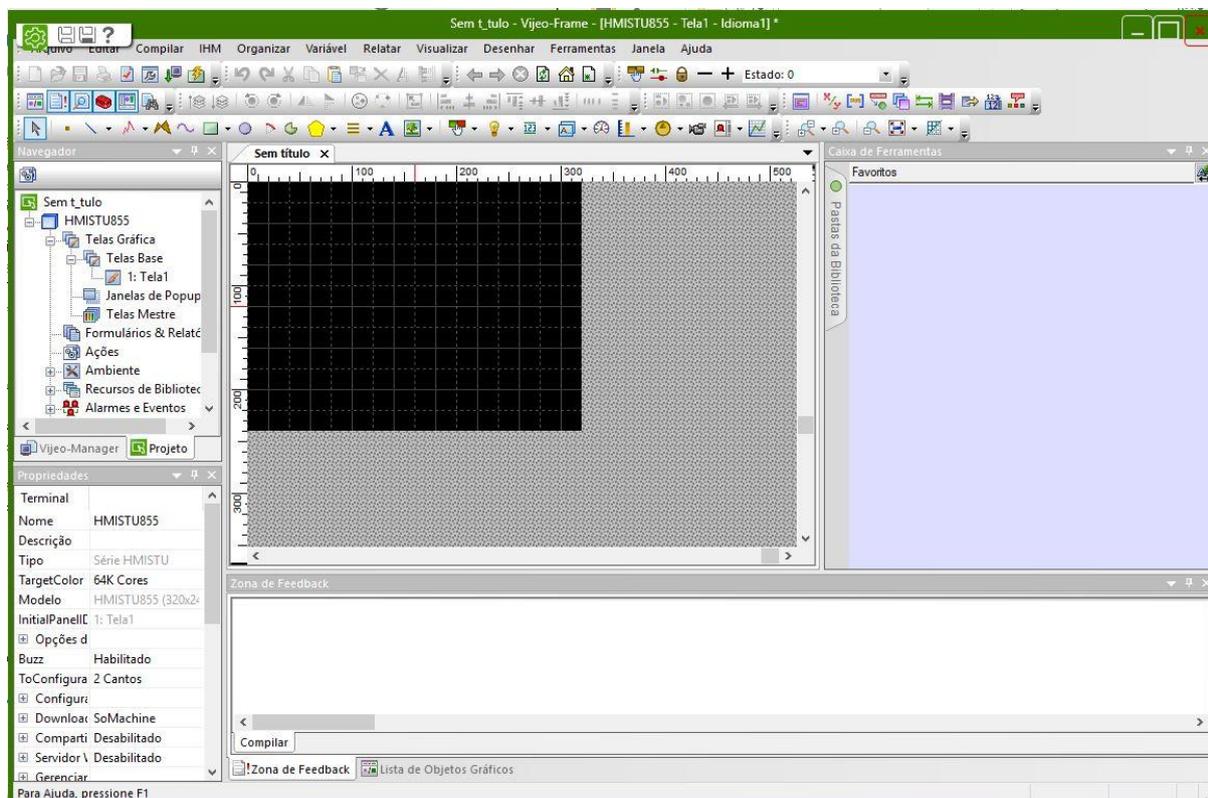


Figura 3.10 - Interface do *Vijeo designer*

3.5 Interface homem-máquina

A fim de que o controle do robô seja visualizado pelo programador, vê-se necessário a instalação de uma interface homem-máquina.

Para este projeto, é utilizado o modelo HMISTU855 da Schneider Electric. As configurações do dispositivo podem ser vistas na figura 3.11.

Pela figura, vê-se que a parte visível ao programador possui uma tela de 129.4x163 mm. Por essa tela é possível fazer a visualização do sistema, bem como fazer seu controle. Pode-se ver também que o HMI possui algumas portas de entrada na parte traseira. Elas são: Porta USB Standard, que faz a conexão do cabo de transferência de dados ou unidade de memória; um Serial I/F, que conecta um cabo serial RS-232C ou RS-485 da CLP ao HMI; uma interface Ethernet, que faz a conexão de um cabo Ethernet à unidade; e um terminal de entrada de energia. A tensão de alimentação do dispositivo é de 24 Vdc e tem uma corrente de partida de 30 ampères.

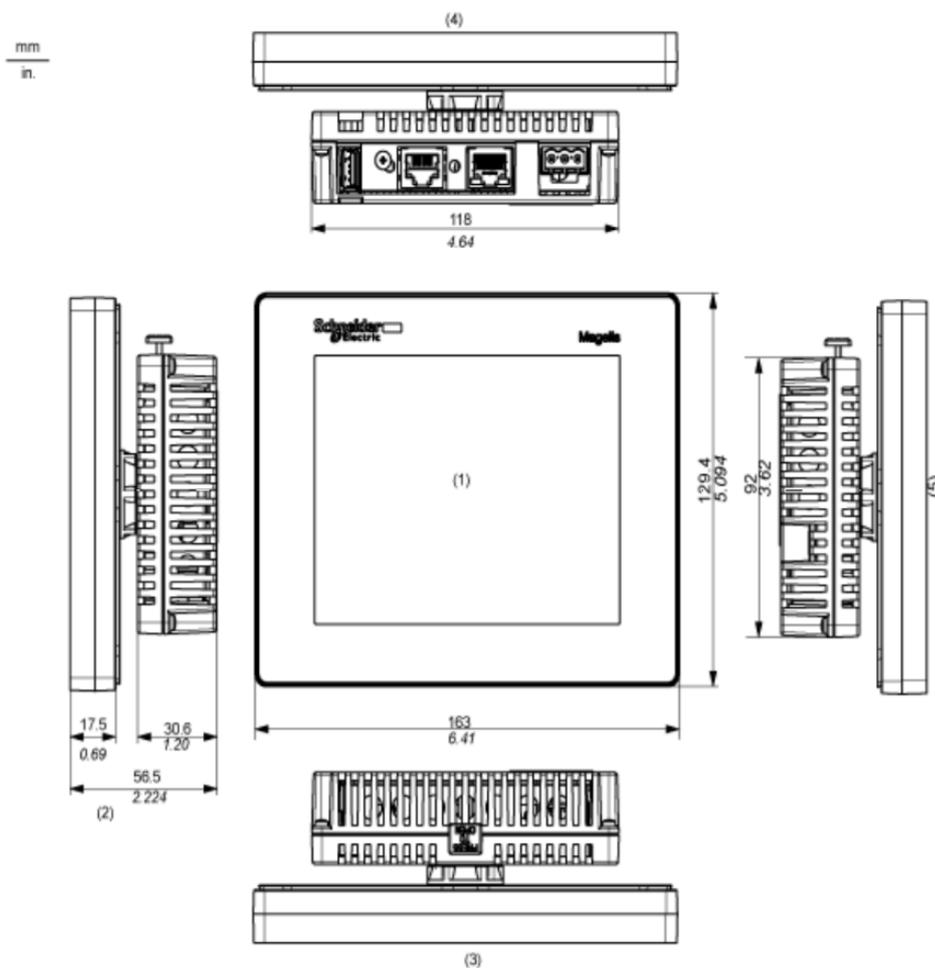


Figura 3.11 - Dimensões do visor do HMISTU855 [27]

3.6 Elementos complementares

Os dispositivos listados nas seções anteriores são utilizados para que seja feito o correto controle do manipulador. Além deles, os dispositivos listados abaixo também foram utilizados para segurança e interligação dos dispositivos e energização dos mesmos.

- Chave liga/desliga: Utilizado para ligar o painel e permitir o fluxo de energia da rede de distribuição aos dispositivos instalados no painel;
- Botão de emergência: Utilizado para desabilitar os movimentos do robô;
- Luzes indicadoras: Três luzes foram selecionadas, uma vermelha para mostrar que o painel está ligado, uma amarela para indicar que existe algum problema e uma verde, que mostra que um processo está em andamento;
- Disjuntores: Dois disjuntores de 6A, para segurança dos drives LXM32MU90M2 e um disjuntor de 10A para segurança do drive LXM32MD18M2;
- Cooler com filtro: Dois coolers com filtro para evitar que o painel superaqueça;
- Fontes de energia: Duas fontes de energia de 24Vdc;

- Sensor de fim de curso: Seis sensores de fim de curso, do tipo sensor de contato, foram instalados no modo NF (normalmente fechado) nos eixos do robô para indicar os pontos iniciais e finais do volume de trabalho. Foram utilizados dois sensores para cada eixo;
- Cabos de potência VW3M5101R50: Cabo de potência para servo-motor, composto por quatro condutores de 1,5 mm², para a energização das bobinas do motor, e dois cabos de 1 mm², para energização do módulo *Holding brake* (freio);
- Cabos de encoder *Schneider Electric* VW3M8102R50: Cabo de encoder para conexão entre os terminais dos servo-motores e servo-drives. A composição do cabo é 3x(2x0.14mm²) + (2x0.34mm²);
- Cabo CANopen *Schneider Electric* VW3M3805R030: Cabo Canopen pré-montado com um terminal fêmea SUB-D 9 e um terminal RJ45;
- Cabo *Allen Bradley Devicenet* 1485C-P1-C: Cabo para conexão dos sensores de fim de curso ao controlador de movimentos, cuja composição é 4x0.75mm²;
- Cabo *Pirelli-Prysmian CORDPLAST* 3x2.5mm² 500V: Cabo flexível cuja composição é 3x2.5mm².

CAPÍTULO 4 – Projeto de Instalação Elétrica

A instalação elétrica do controlador de movimentos e dos servo-drives foi feita de acordo com os manuais de instrução disponibilizados pela Schneider Electric, bem como de acordo com as normas NBR5410/2004, que constitui um guia normativo para o dimensionamento de condutores, de proteção contra sobrecorrentes, de circuitos e instalações de motores. Essa norma, vista na figura 4.1, fala sobre o dimensionamento dos condutores com base nas capacidades de condução de corrente, em ampères, para os métodos de referência A1, A2, B1, B2, C e D. O método de referência escolhido foi o B1, método de instalação indicado na IEC 364-5-523 para os quais a capacidade de condução de corrente foi determinada por meio de ensaio ou de cálculo. O eletroduto escolhido possui três condutores (fase-neutro-terra) e o material do condutor é o cobre. O modo B1 diz respeito a condutores isolados instalados em eletroduto de seção circular sobre parede de madeira.

Seções nominais mm ²	Métodos de instalação definidos na tabela 28											
	A1		A2		B1		B2		C		D	
	2 condutores carregados	3 condutores carregados	2 condutores carregados	3 condutores carregados	2 condutores carregados	3 condutores carregados	2 condutores carregados	3 condutores carregados	2 condutores carregados	3 condutores carregados	2 condutores carregados	3 condutores carregados
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
Cobre												
0,5	7	7	7	7	9	8	9	8	10	9	12	10
0,75	9	9	9	9	11	10	11	10	13	11	15	12
1	11	10	11	10	14	12	13	12	15	14	18	15
1,5	14,5	13,5	14	13	17,5	15,5	16,5	15	19,5	17,5	22	18
2,5	19,5	18	18,5	17,5	24	21	23	20	27	24	29	24
4	26	24	25	23	32	28	30	27	36	32	38	31
6	34	31	32	29	41	36	38	34	46	41	47	39
10	46	42	43	39	57	50	52	46	63	57	63	52
16	61	56	57	52	76	68	69	62	85	76	81	67
25	80	73	75	68	101	89	90	80	112	96	104	86
35	99	89	92	83	125	110	111	99	138	119	125	103
50	119	108	110	99	151	134	133	118	168	144	148	122
70	151	136	139	125	192	171	168	149	213	184	183	151
95	182	164	167	150	232	207	201	179	258	223	216	179
120	210	188	192	172	269	239	232	206	299	259	246	203
150	240	216	219	196	309	275	265	236	344	299	278	230
185	273	245	248	223	353	314	300	268	392	341	312	258
240	321	286	291	261	415	370	351	313	461	403	361	297
300	367	328	334	298	477	426	401	358	530	464	408	336
400	438	390	398	355	571	510	477	425	634	557	478	394
500	502	447	456	406	656	587	545	486	729	642	540	445
630	578	514	526	467	758	678	626	559	843	743	614	506
800	669	593	609	540	881	788	723	645	978	865	700	577
1 000	767	679	698	618	1 012	906	827	738	1 125	996	792	652

Figura 4.1 - Capacidades de condução de correntes, em ampères, para os métodos de referência A1, A2, B1, B2, C e D para o cobre [28]

4.1 Alimentação do painel de controle e distribuição de energia para os dispositivos

A escolha do cabo utilizado para fazer a alimentação do painel de controle foi tomada com base na figura 4.1 e nas correntes de pico de saída dos servo-drives e das fontes instaladas no painel. A corrente de entrada dos drives LXM32MU90M2 é de 4,5 A, do drive LXM32MD18M2, 8,4 A, e das fontes é de 500 mA. Assim, o cabo de alimentação deve ter capacidade de transportar correntes de, pelo menos, 18,4 A.

A alimentação do painel de controle é feita a partir do quadro de distribuição geral do Graco, que possui tensão de 220V. Como o cabo escolhido deve ter capacidade para transportar pelo menos 18,4 A, com base na figura 4.1, o cabo deve possuir bitola de pelo menos 2,5 mm², o que limita a corrente a 21 A. Assim, foi escolhido um cabo *Prysmian-Pirelli* composto por cabos internos de 2.5 mm², sendo eles:

- Marrom: Fase;
- Azul: Neutro;
- Preto: Terra.

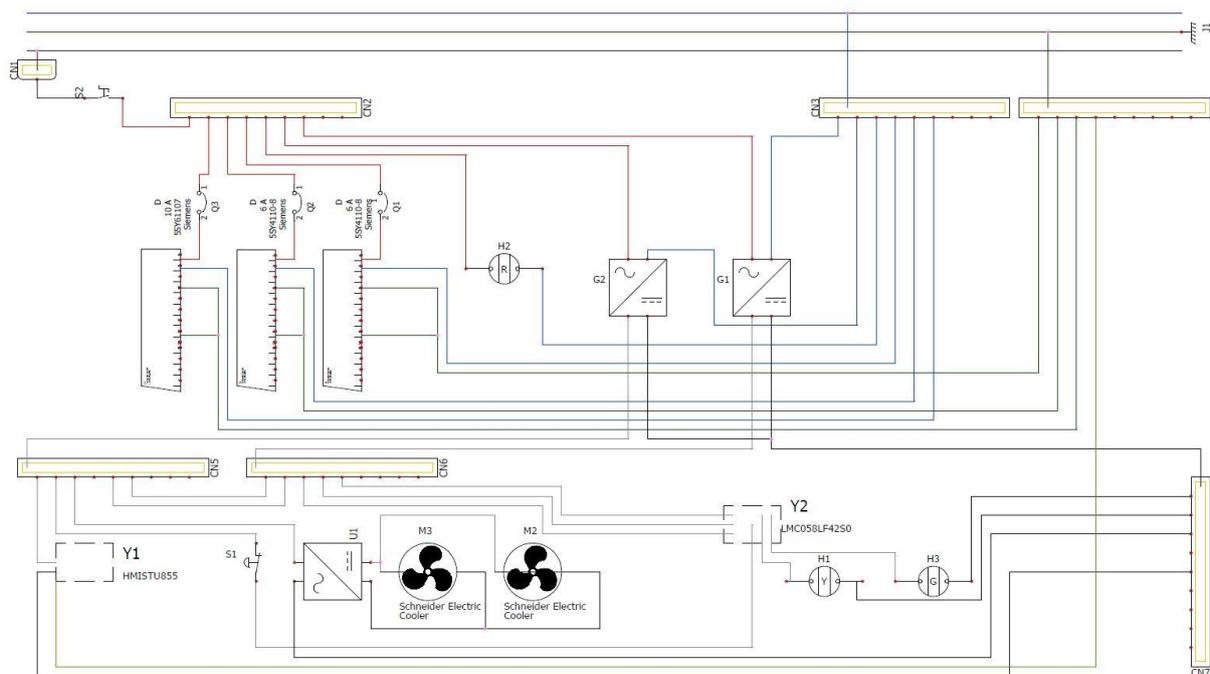


Figura 4.2 - Esquema de fiação dos dispositivos (Ver Apêndice III)

O painel é ligado utilizando um botão liga/desliga. Quando esse botão é habilitado, ocorre, então, o fluxo de energia do quadro geral para os dispositivos dentro do painel. Assim, o cabo fase que vem do quadro geral de distribuição passa primeiramente por um único borne que vai direto pra esse botão, e é então levado aos outros bornes de distribuição por meio de um cabo flexível vermelho de 2.5mm². A energia é levada por cabos flexíveis vermelhos de 2.5mm² aos servo-drives, por um cabo cinza de 1mm² às fontes de 24Vdc e à luz indicadora vermelha, que são os únicos elementos ligados diretamente ao quadro geral de distribuição. A luz indicadora vermelha terá, então, a função de mostrar que o painel está ligado.

As fontes de 24Vdc têm a função de alimentar o controlador de movimentos, a conexão CN2 dos servo-drives, o HMI, os *coolers* e os sensores de fim de curso instalados em cada articulação do robô. Os *coolers* são alimentados com tensão de 12 Vdc, assim decidiu-se utilizar um dispositivo redutor de tensão entre a fonte e os *coolers* para fazer sua correta alimentação. Além disso, um cabo flexível cinza de 1mm² foi utilizado para passar pelo botão de emergência para enviar um sinal direto à primeira conexão do módulo 51 do controlador de movimentos e aos módulos CN2 dos drives. Assim, quando o botão de emergência for pressionado, a tensão que é levada à conexão será cortada, e informará que o botão foi pressionado e a movimentação do robô será desabilitado. Para os coolers, como sua tensão de alimentação é de 12V, foi utilizado um dispositivo de redução de tensão.

O esquema de alimentação dos dispositivos do painel pode ser visto na figura 4.2. O desenho técnico da instalação elétrica pode ser visto de maneira mais detalhada no Apêndice III. Para a distribuição de energia para os dispositivos, foram utilizados módulos de borne, e, para que os cabos não ficassem à mostra, foram utilizados módulos de passa-cabos. A figura 4.2 mostra um esquema simplificado apenas para ter uma base de como foi feita a distribuição dos fios para alimentação dos dispositivos.

O esquema de distribuição dos dispositivos dentro do painel pode ser visto nas figuras 4.3 e 4.4. Os desenhos técnicos para a furação da caixa de montagem e de sua tampa podem ser vistos no Apêndice II.

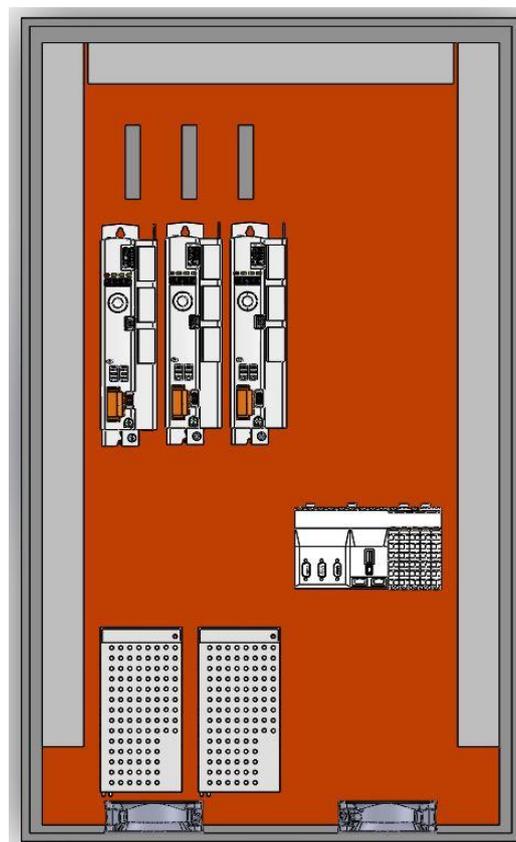


Figura 4.3 - Organização final dos dispositivos dentro do painel de controle

O módulo de passa-cabos localizado à esquerda foi utilizado para os fios que transportam cargas de 24 V. O passa-cabos localizado à direita e na parte superior do painel foi designado para alocar os fios que transportam as tensões de 220V, quem vêm direto no quadro geral de distribuição de energia.

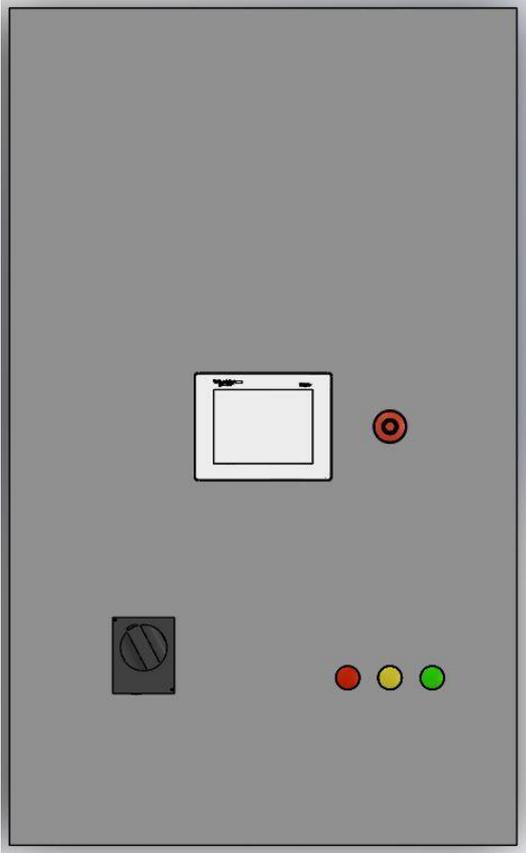


Figura 4.4 - Arranjo dos dispositivos na tampa do painel de controle

A figura 4.5 mostra de maneira simplificada a interconexão entre os dispositivos do painel e dos servo-drives com os servo-motores.

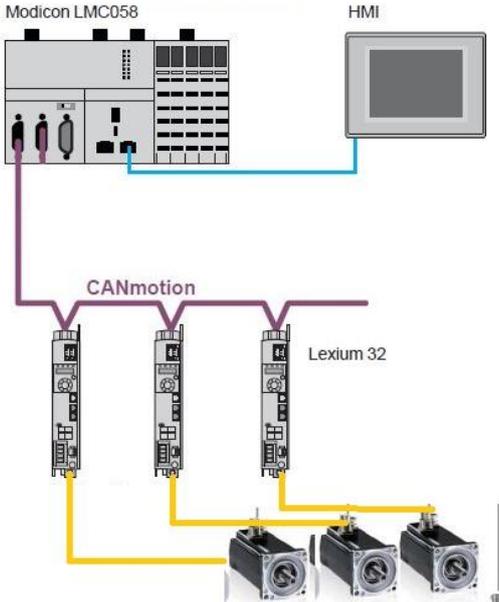


Figura 4.5 - Interconexão dos dispositivos

4.2 Instalação elétrica dos Servo-drives

Para fazer a conexão dos servo-drives com a energia, ou seja, sua alimentação, é preciso tomar nota, primeiramente, de que o produto tem uma corrente de fuga maior que 3,5 mA. Isso significa que, caso o dispositivo não seja aterrado corretamente, podem ocorrer danos aos outros produtos localizados no painel de controle bem como pode existir o risco de choque elétrico caso o usuário venha a tocar no produto.

O manual do usuário explicita alguns cuidados que o técnico ou instalador deve ter antes de realizar a alimentação dos servo-drives, que, caso não sejam seguidas, podem provocar riscos ao técnico. Assim, antes de ligar o dispositivo na energia, é preciso fazer sua correta conexão ao terminal terra do painel. O condutor de proteção deve ter, pelo menos, 10 mm² de seção transversal e não pode ser utilizado cabo blindado para fazer a conexão do dispositivo com o terminal terra.

O manual de instalação dos servo-drives prevê que sua instalação ocorrerá em indústrias ou áreas em que a corrente que chegará ao dispositivo será muito alta. Porém, como ele será instalado no painel do GRACO, pode-se, então, utilizar um cabo de aterramento com seção transversal menor, já que as tarefas que o robô irá executar não necessitarão de potência muito elevada.

Assim, o aterramento dos servo-drives foi feito utilizando um cabo flexível verde de 4mm² na conexão de terra do dispositivo, localizado em sua parte inferior, como pode ser visto na figura 4.6. Os cabos foram então conectados ao terminal geral de aterramento do painel.

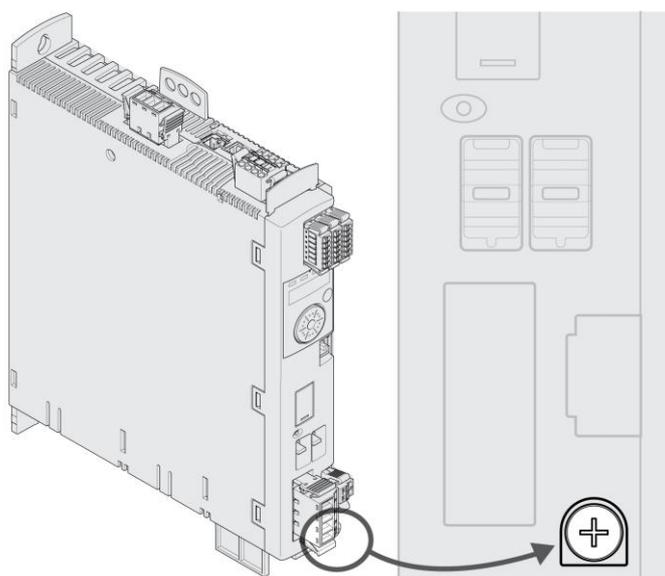


Figura 4.6 - Parafuso para aterramento do servo-drive [14]

Como os servo-drives devem ser alimentados com tensão de 220V, eles foram conectados à energia diretamente, através do cabo de alimentação que vem do quadro geral de distribuição de energia. Assim, deve-se ter conhecimento de como deve ser a especificação do cabo para fazer a alimentação correta dos servo-drives. O manual explicita que a seção transversal do cabo utilizado para fazer a alimentação dos servo-drives deve possuir seção transversal suficientemente grande para que possa ser deslocado caso seja

necessário, e essa seção transversal deve ser entre 0.75 mm² e 5.3 mm². A visão geral de alimentação dos dispositivos pode ser vista na figura 4.7.

Com base na corrente de pico de nove ampères, para os servo-drives dos eixos Y e Z, e de 18 ampères para o servo-drive do eixo X, viu-se que cabos com bitola de 2.5 mm² é suficiente para fazer a correta alimentação dos dispositivos. Assim, foram escolhidos os cabos:

- Vermelho: Fase;
- Azul: Neutro;
- Verde e amarelo: Terra

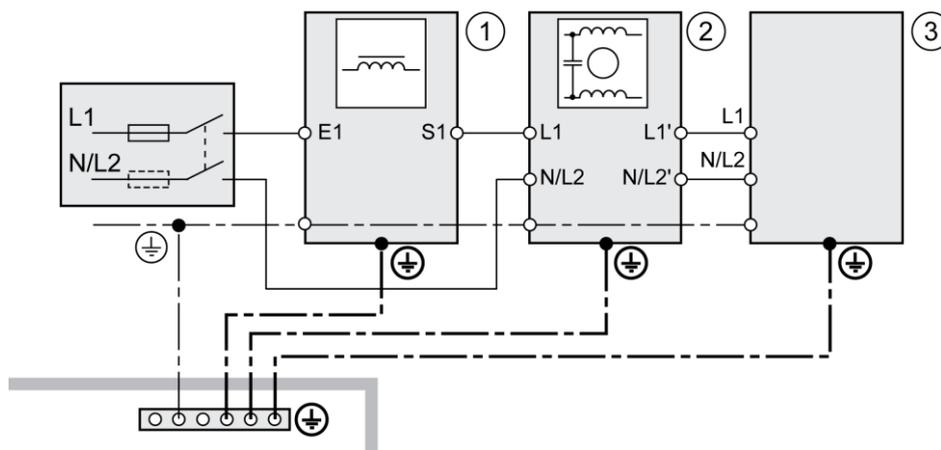


Figura 4.7 - Visão geral da alimentação para o dispositivo [14]

A conexão foi feita na entrada CN1, visto na figura 4.8, em seus respectivos *slots*, e, para proteção contra sobrecorrentes, foram utilizados três disjuntores, sendo que para o LXM32MD18M2 foi utilizado um disjuntor de 10A e para os servos LXM32MU90M2 foram utilizados dois disjuntores de 6A cada. Na figura 4.7, 1 representa o disjuntor, 2 representa um acessório de cabeamento (não utilizado) e 3 representa o servo-drive.

CN1 Mains 115/230 Vac

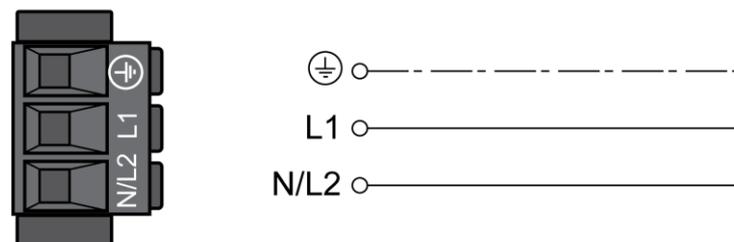


Figura 4.8 - Diagrama de cabeamento dos servo-drives [14]

O símbolo terra na figura 4.8 representa o cabo verde-amarelo, o cabo L1 representa o fio vermelho (fase) e o cabo N/L2 representa o fio azul (neutro).

Ao módulo CN2 foram conectados cabos de 1mm² cinza às entradas STO_A, STO_B, 24V e 0V, de acordo com a figura 4.9.

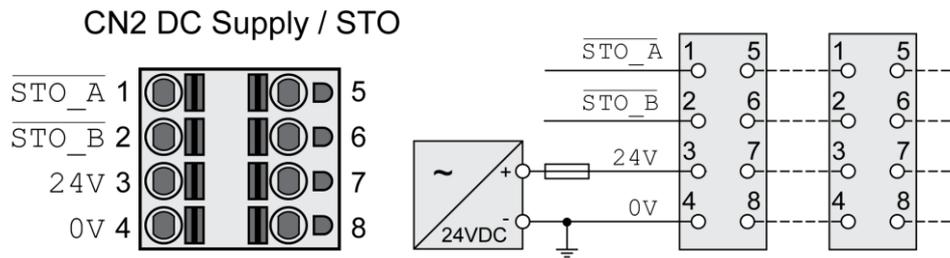


Figura 4.9 - Módulo CN2 do servo-drive [14]

As entradas 24V e 0V são entradas de alimentação do módulo, e tem energia fornecida pelas fontes instaladas no painel. As entradas STO_A e STO_B têm função de *Safe Torque Off*, ou seja, elas inibem o funcionamento do drive quando uma energia deixa de ser enviada a essas entradas e param o movimento dos servo-motores. Para seu funcionamento, dois cabos cinza foram conectados partindo do botão de emergência (NF), que alimenta essas entradas. Assim, quando o botão de emergência é pressionado, a energia que alimenta essas entradas é cortada e, por consequência, os drives são desabilitados imediatamente.

4.3 Instalação elétrica do Controlador de Movimentos

O módulo de controle e distribuição de energia (CPDM) do controlador de movimentos possui três conexões de energia:

- *Main Power*,
- *Expert I/O*;
- *24 Vdc I/O Power Segment power*.

O *Main Power* alimenta o módulo TM5, ou seja, o próprio controlador de movimentos, além de fornecer energia para a porta *Serial Line*, porta USB e qualquer módulo PCI que possa vir a ser conectado ao controlador. O *Expert I/O Power* alimenta os módulos 41 e 42, além de alimentar o módulo embarcado do encoder. Já o *24 Vdc I/O Power Segment power* alimenta os módulos regulares 51 e 52.

Na figura 4.10 é mostrada a tela onde estão localizadas luzes indicadoras que identificam se o controlador está sendo alimentado ou não. A tabela 4.1 descreve o comportamento dessas luzes indicadoras.

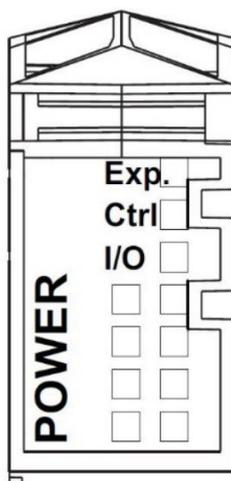


Figura 4.10 - Luzes indicadoras do controlador de movimentos [12]

Luz indicadora	Cor	Status	Descrição
Exp	Verde	Ligado	Tensão 24 Vdc aplicada
Ctrl	Verde	Ligado	Tensão 24 Vdc aplicada
I/O	Verde	Ligado	Tensão 24 Vdc aplicada

Tabela 4.1 - Descrição das luzes indicadoras do controlador de movimentos

O controlador de movimentos deve ser alimentado com tensão de 24V. Além disso, a corrente máxima de alimentação de cada um dos módulos citados acima deve ser de:

- *Main Power*: 0,6 A;
- *Expert I/O*: 1,15 A;
- *24 Vdc I/O Power Segment power*: 6,3 A;

Assim, para sua energização, foram utilizadas duas fontes, que foram conectadas ao seu módulo 31. Foram utilizados três cabos flexíveis cinzas de 1mm² e três cabos flexíveis pretos de 1mm², sendo:

- Cinza: 24V;
- Preto: 0V

Para proteção do controlador de movimentos, foram usados três fusíveis de 2A cada para as conexões. Na figura 4.11 é mostrado como a conexão é feita

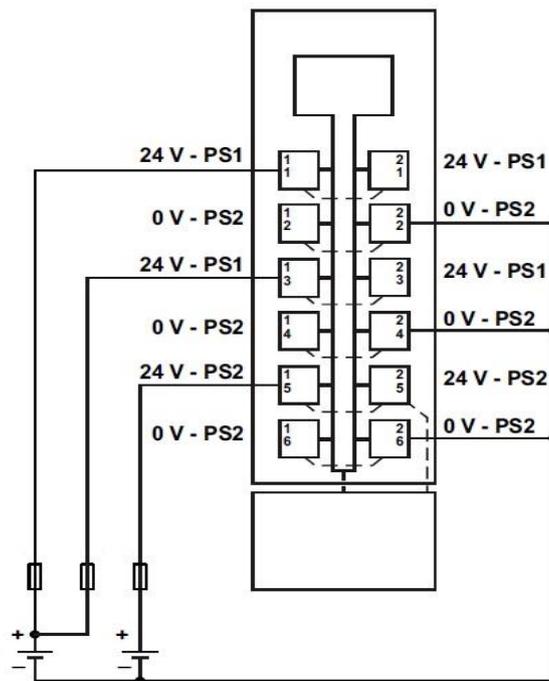


Figura 4.11 - Diagrama de alimentação do controlador de movimentos [12]

O botão de emergência é também conectado ao módulo 51 do controlador de movimentos. Um fio de 1mm² sai do borne alimentado pela fonte, passa pelo botão de emergência e chega ao controlador de movimentos. Quando o botão é acionado, a tensão que entra no controlador vai para zero e ele envia um comando aos drives para que a movimentação seja abortada e os drives são desabilitados.

Do módulo 52 do LMC058, saem dois fios de 1mm² que alimentam as luzes indicadoras verde e amarela. Quando um processo está ocorrendo, a luz verde é acionada e, caso haja algum problema com o processo, o controlador aciona a luz amarela.

Para conectar o computador ao controlador de movimentos, é utilizado um cabo *Ethernet* com dois terminais RJ45.

4.4 Instalação elétrica da Interface Homem-Máquina

A alimentação do dispositivo HMI também deve ser feita com 24V. Assim, para sua energização foi utilizada uma fonte 24Vdc. Foram utilizados três cabos flexíveis de 1mm, sendo:

- Cinza: terminal positivo 24 V;
- Preto: terminal negativo 0 V;
- Verde: terra, terminal FG

Esses fios são colocados no plug de conexão à entrada de alimentação da interface homem-máquina, conforme pode ser visto na figura 4.12.

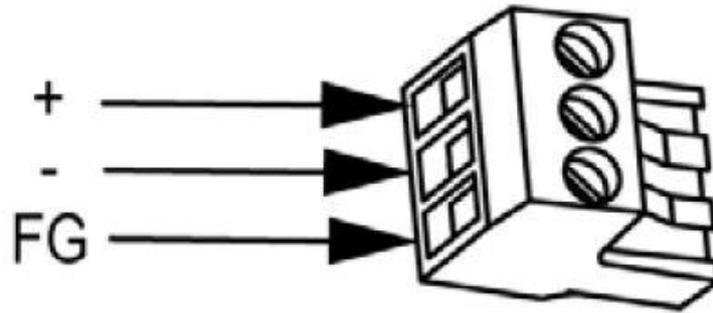


Figura 4.12 - Ilustração do *plug* do HMI [27]

O terminal para conexão da fonte de energia com o dispositivo fica localizado em sua parte inferior, como pode ser visto na figura 4.13.

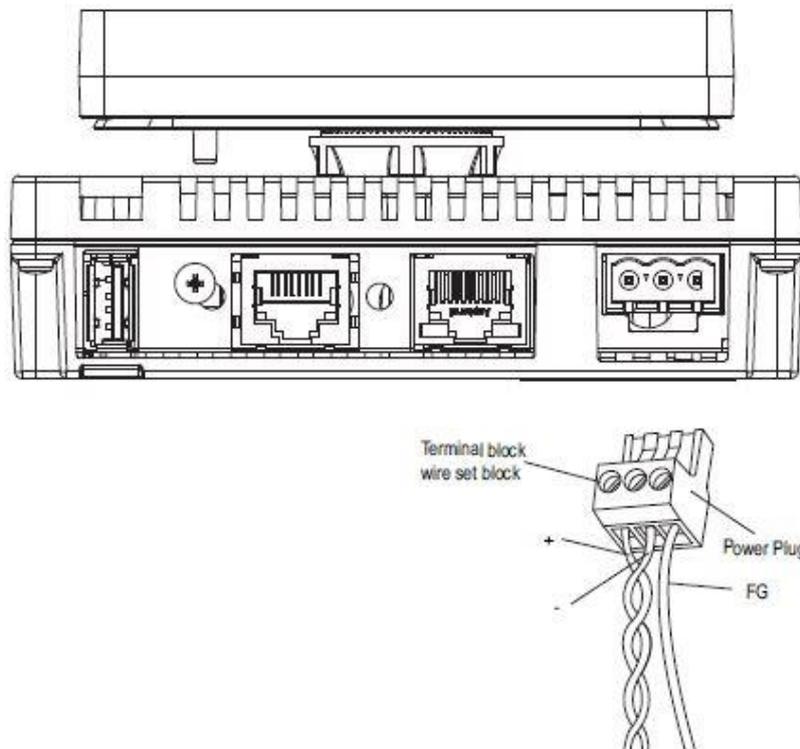


Figura 4.13 - Conexão do plug ao HMI [27]

A conexão entre o controlador de movimentos e o HMI é feito por um cabo flexível do tipo USB-USB mini, que conecta a entrada USB do HMI à entrada USB mini do LMC058.

4.5 Instalação elétrica dos servo-motores

Como dito no capítulo anterior, os servo-motores não podem ser alimentados diretamente da rede geral de energia. Por esse motivo eles são, então, alimentados pelos servo-drives instalados no painel de controle.

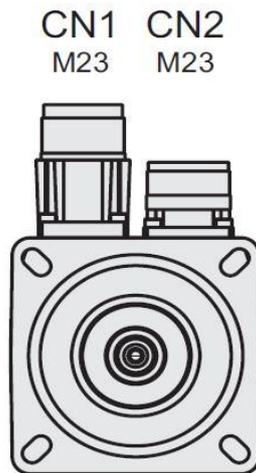


Figura 4.14 - Modelo para as entradas de alimentação do servo-motor [8]

Na figura 4.14 é possível ver os slots em que será conectado o cabo para alimentação do motor. No caso é utilizada a entrada CN1. A figura 4.15 mostra a atribuição para os pinos em que será conectado o cabo para energização do servo-motor. A tabela 4.2 explica a função desses pinos.

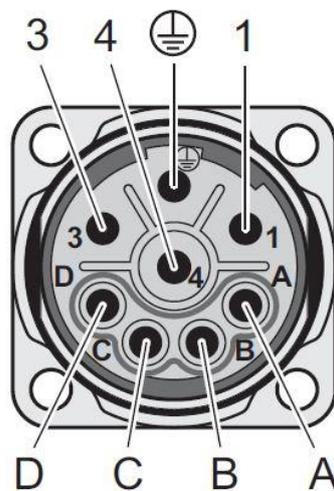


Figura 4.15 - Pinos de conexão do cabo de alimentação do servo-motor [8]

Pino	Atribuição	Significado
1	U	<i>Motor phase U</i>
	PE	Condutor protetor para o terra
3	W	<i>Motor phase W</i>
4	V	<i>Motor phase V</i>
A	BR+	Fonte de tensão para o <i> Holding brake 24Vdc</i>
B	BR-	Fonte de tensão para o <i> Holding brake 0Vdc</i>
C	Reservado	-

D	Reservado	-
	SHLD	Proteção para invólucro do servo-motor

Tabela 4.2 - Descrição dos pinos de entrada do servo-motor

Para que o cabo de energização seja conectado corretamente à entrada CN1 do servo-motor, ele deve ser montado de acordo com a figura 4.16. Como pode ser visto das figuras 4.15 e 4.16, o cabo deve ser composto por seis fios internos que fazem a alimentação do motor e sua proteção. São eles:

- BK U1: para alimentação do pino *Motor phase U*;
- BK V2: para alimentação do pino *Motor phase V*;
- BK W3: para alimentação do pino *Motor phase W*;
- GN/YE: Condutor protetor para o terra;
- GY: Condutor para o *Holding brake 24 Vdc*;
- WH: Condutor para o *Holding brake 0 Vdc*.

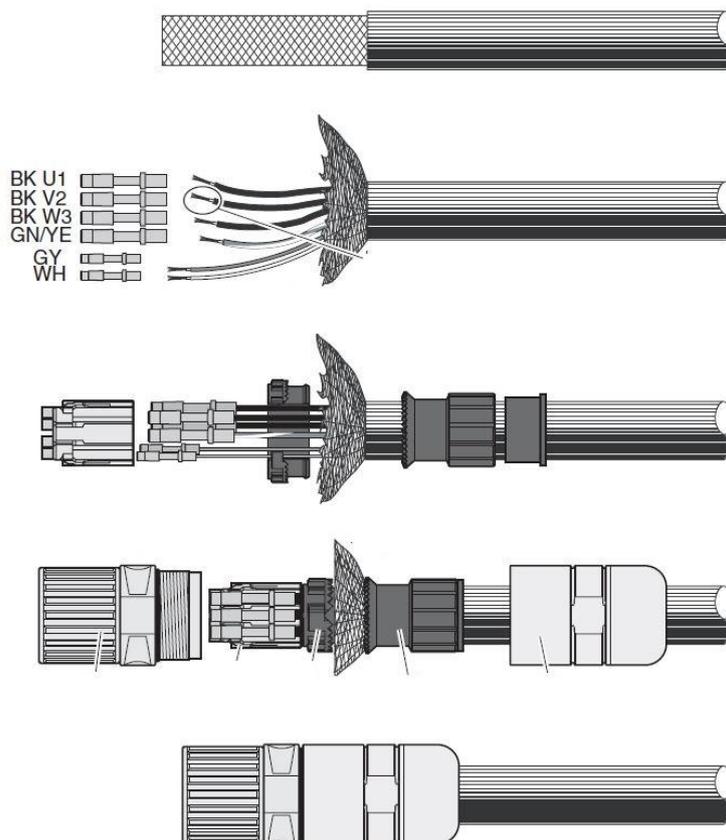


Figura 4.16 - Configuração do cabo de alimentação do servo-motor [8]

O manual dos servo-drives também fornece informações de como deve ser o cabo que fará a ligação entre os terminais de conexão do drive e dos servo-motores. Primeiramente são fornecidas as especificações dos cabos, que podem ser vistas na tabela 4.3.

Proteção	Necessário ter os dois terminais aterrados
PELV	Necessário que os fios de conexão com os terminais Holding brake sejam do tipo PELV
Composição do cabo	3 fios – Motor phases; 2 fios – Holding brake; Os condutores devem ter seção transversal grande o suficiente para que possam ser deslocados caso seja necessário;
Tamanho máximo do cabo	Depende da utilização do servo-drive

Tabela 4.3 - Especificações do cabo de alimentação dos servo-motores

Além das especificações citadas na tabela 4.3, os fios que compõem o cabo deve ter seção transversal específica. Referente às entradas dos servo-drives, para a conexão com o terminal CN10, os fios devem possuir seção transversal entre 0,75 e 5,3 mm², enquanto para a conexão com o terminal CN11, os fios devem possuir seção transversal entre 0,75 e 2,5 mm². A especificação em relação à proteção explicita que é necessário que exista um fio de proteção na composição do cabo, além de ser do tipo PELV, ou seja, que tenha o sistema de proteção de extra baixa tensão. A figura 4.17 mostra como deve ser a configuração do cabo.

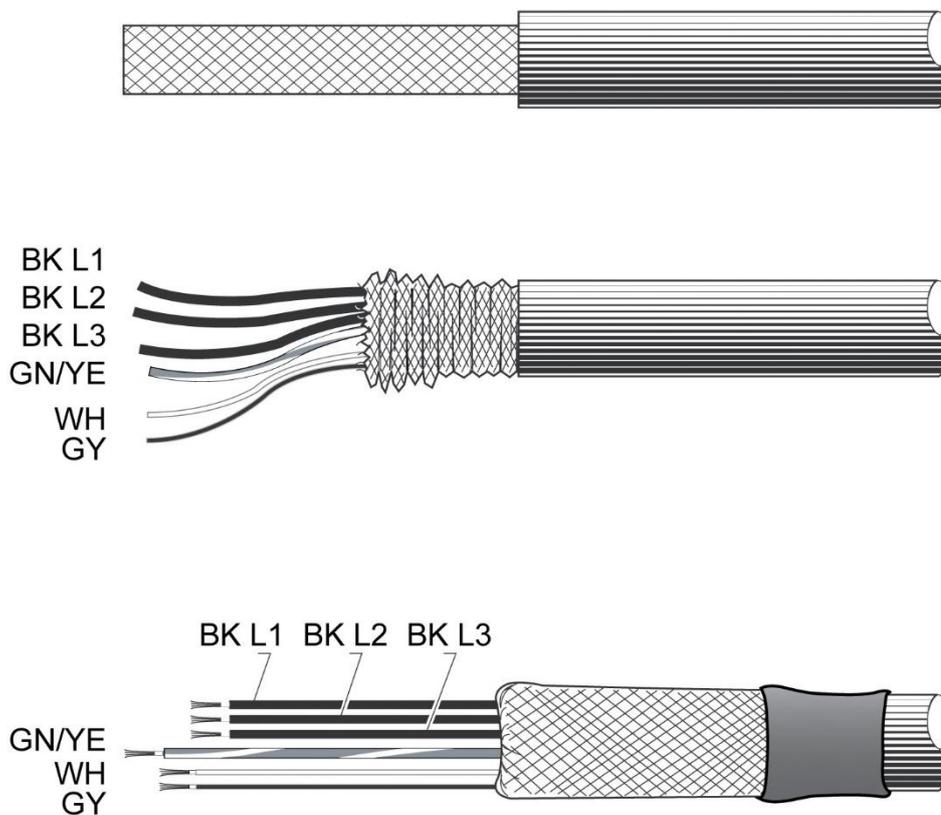


Figura 4.17 - Composição do cabo para conexão com os terminais CN10 e CN11 [14]

Então, para fazer a correta alimentação dos servo-motores e sua conexão com seus respectivos servo-drives, foram utilizados cabos de potência VW3M5101R050, da Schneider Electric. Esse cabo é conectado do terminal do servo-motor à conexão CN10 do servo-drive e um dos terminais já vem montado de acordo com as especificações mostradas na figura 4.16, para conexão com o terminal CN1 do servo-motor, e tabela 4.2. Esse cabo é composto por seis fios, sendo quatro fios de 1,5mm² e dois fios de 1 mm². Os fios de 1,5 mm² são divididos em:

- Black L1: Para entrada *Motor phase U*;
- Black L2: Para entrada *Motor phase V*;
- Black L3: Para entrada *Motor phase W*;
- Verde/amarelo: Conductor protetor para aterramento;

Esses fios são utilizados para fazer a energização do servo motor e são conectados ao módulo CN10 do servo-drive correspondente, conforme visto na figura 4.18.

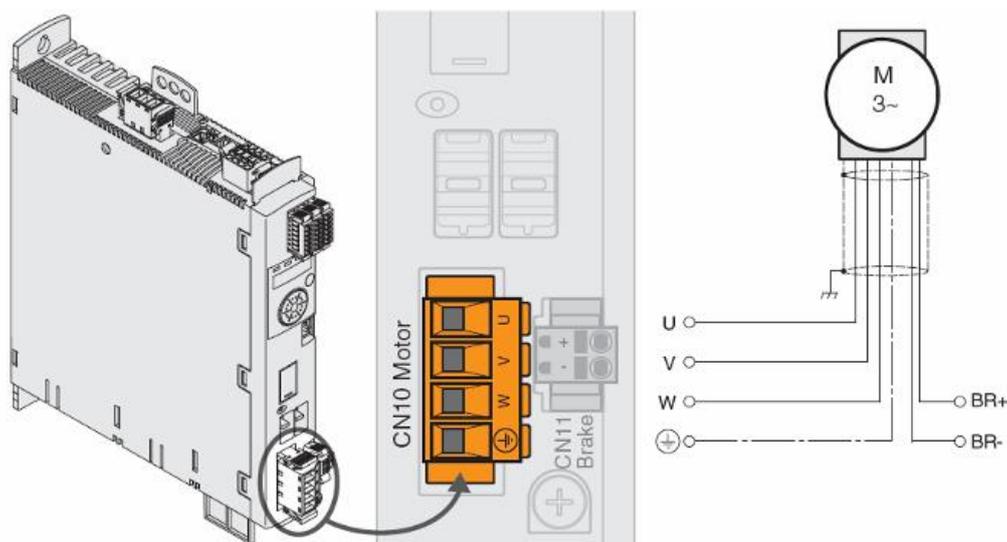


Figura 4.18 - Conexão do cabo de alimentação ao terminal do servo-drive [29]

Os dois cabos de 1mm² foram conectados ao terminal CN11, como mostrado na figura 4.19, e são divididos em:

- Branco: *Holding brake* entrada positiva;
- Cinza: *Holding brake* entrada negativa;

Esses dois fios têm a função de alimentar os terminais *Holding brake* 24 Vdc e 0 Vdc, respectivamente, dos servo-motores. Esses terminais têm a função de travar o eixo para que ele não faça movimentações indesejadas.

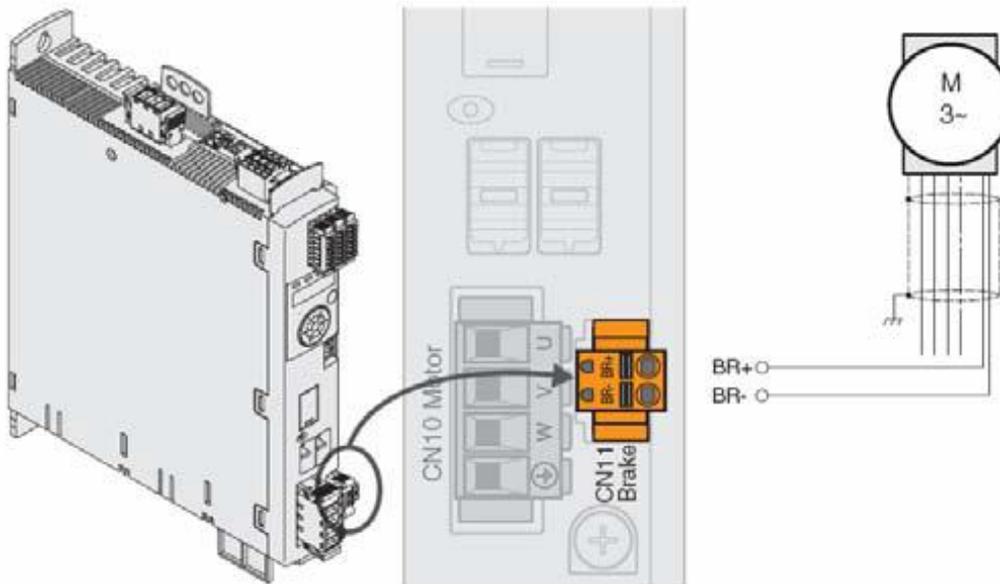


Figura 4.19 - Conexão para o terminal *Holding brake* [29]

Esse cabo tem, assim, a função de alimentar os servo-motores e de dar proteção. A conexão com os terminais CN10 e CN11 dos servo-drives é feita conforme mostra a figura 4.20

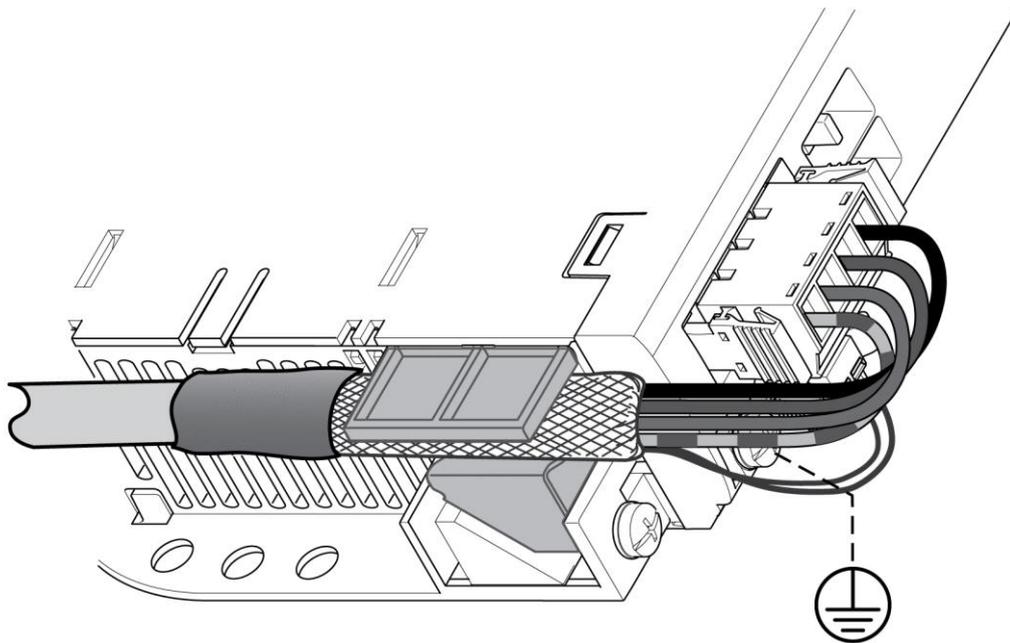


Figura 4.20 - Vista da conexão do cabo de energização dos motores aos módulos CN10 e CN11 [14]

4.6 Alimentação dos sensores de fim de curso e conexão com os servo-drives

Os eixos do robô também são conectados aos drives por meio de sensores de fim de curso, do tipo sensor de contato com conexão NF. Em cada eixo do manipulador foram colocados dois sensores, que indicam o ponto inicial e o ponto final da movimentação do braço robótico. A alimentação dos sensores é

feita por meio da fonte de energia localizada no painel. O cabo escolhido para fazer a ligação entre os sensores e os drives e alimentação dos sensores foi o *Allen Bradley Devicenet 1485C-P1-C*. Este cabo foi utilizado por estar disponível no laboratório e atender às necessidades de conexão dos sensores de contato aos servo-drives. Este cabo possui quatro fios de 0.75mm², sendo:

- Vermelho: alimentação positivo;
- Preto: alimentação negativo;
- Azul: Informação;
- Branco: Informação;

Assim, o fio vermelho parte do borne alimentado pela fonte, e alimenta o sensor. Quando a articulação do eixo atinge o ponto inicial, a tensão enviada pelo fio azul à entrada DI3 do drive correspondente ao eixo é interrompida, para informar ao drive que a posição inicial foi atingida. Quando a articulação do eixo atinge o ponto final de atuação do eixo, a tensão enviada para a entrada DI2 é interrompida, para informar que o ponto final foi atingido. Essas entradas encontram-se no módulo CN6. Para o correto funcionamento desse módulo, foi também conectado um fio cinza 1mm² da entrada DQCOM do módulo CN6 à entrada 0V do módulo CN2. O módulo CN6 pode ser visto na figura 4.21.

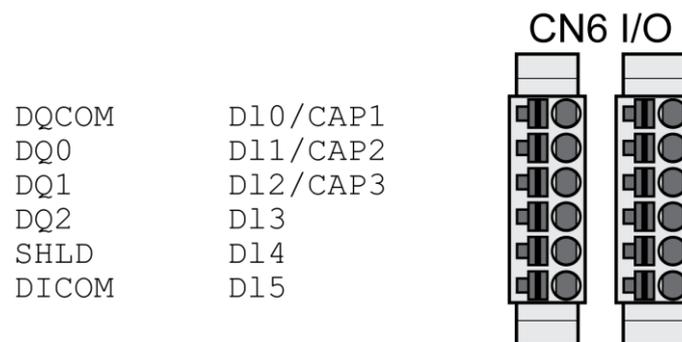


Figura 4.21 - Módulo CN6 [14]

CAPÍTULO 5 – Configuração dos Dispositivos e Programação do Controlador de Movimentos

A programação do LMC058 para movimentação dos eixos do manipulador foi dividida em quatro etapas, sendo elas:

- Comissionamento dos drives e comunicação com o controlador de movimentos;
- Teste de locomoção dos eixos;
- Confecção do algoritmo para programação;
- Realização do programa e testes;

As etapas de comissionamento e teste de locomoção foram feitas baseadas nos manuais dos drives, do robô e dos servo-motores utilizados no projeto. Já a confecção do algoritmo e realização do programa foram baseados em programas-exemplo encontrados no próprio software Somachine V4.1.

5.1 Comissionamento dos drives e comunicação com o controlador de movimentos

Entre os terminais dos servo-motores e servo-drives foram utilizados cabos de encoder VW3M8102R50, cuja função é mandar as informações do encoder do servo-motor para a entrada CN3 dos servo-drives, vista na figura 5.1. O cabo escolhido preenche os pré-requisitos de especificações de cabo descritos no manual do usuário do servo-drive, que podem ser vistos na tabela 5.1, além de já vir montado, o que reduz os riscos de erro na fiação.

CN3 Encoder-1

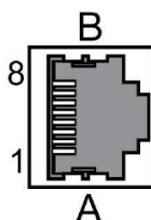


Figura 5.1 - Módulo CN3 [14]

Proteção	Terminais de ambos os lados devem ser aterrados
Par trançado	Necessário
PELV	Necessário
Composição do cabo	6x0,14 mm ² + 2x0,34 mm ²
Tamanho máximo do cabo	100 m

Tabela 5.1 - Especificações do cabo de encoder

O encoder localizado no servo-motor é do tipo absoluto e incremental. Inicialmente, ao ser ligado, é obtido o valor absoluto, medido a partir de uma revolução completa no início da movimentação e, depois, continua a contar os valores de forma incremental após a primeira revolução. Suas principais características podem ser vistas na tabela 5.2. A conexão do cabo de encoder é feita no servo-motor na entrada CN2, visto na figura 4.14.

Resolução por revolução	16 períodos de sen/cos
Faixa de medição absoluta	1 revolução
Acurácia	+/- 0.08°
Forma do sinal	Senoidal
Tensão de alimentação	7-12 Vdc
Corrente máxima de alimentação	50 mA

Tabela 5.2 - Características do encoder do servo-motor

Para a conexão entre o controlador de movimentos e o drive do eixo Z, foi utilizado o cabo CANopen VW3M3805R030. Este cabo é conectado ao terminal de comunicação CANopen que foi instalado ao *slot* 3 do servo-drive. O terminal fêmea SUB-D 9 é conectado à entrada CAN1 do controlador de movimentos, figura 5.2, e o terminal RJ45 é conectado à entrada CAN do *slot* 3 do drive LXM32MU90M2. Os outros dois drives são, então, conectados em cadeia utilizando o mesmo cabo, porém com dois terminais RJ45. Na saída do *slot* 3 do último drive da cadeia foi colocado um terminal RJ45, para fechar a saída.

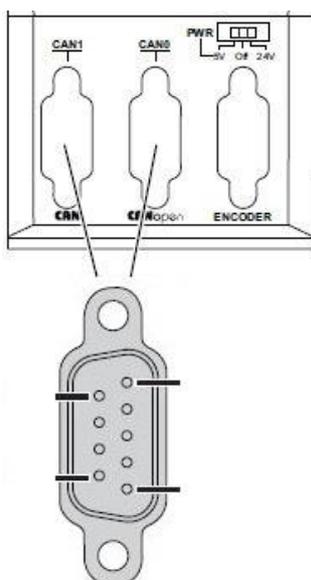


Figura 5.2 - Porta CAN [29 e 14 (Modificado)]

No software Somachine V4.1, primeiramente o usuário deve escolher o dispositivo que será programado. No caso do presente projeto, o controlador de movimentos LMC058S042 é o dispositivo a

ser programado. Por se tratar de um projeto em que eixos de um robô serão movimentados, a entrada CAN1 do controlador de movimentos deverá ser utilizada, já que é a porta que se utiliza para a configuração *Motion Controller*. Essa entrada deverá ter sua configuração de taxa de transmissão feita, bem como sua quantidade de nós. A escolha da taxa de transferência de dados é dependente do tamanho do cabo de conexão, e pode ser vista na tabela 5.3.

Taxa de transferência de dados	1 Mbit/s	800 Kbit/s	500 Kbit/s	250 Kbit/s	125 Kbit/s	50 Kbit/s	20 Kbit/s	10 Kbit/s
Tamanho máximo do cabo	4 m	25 m	100 m	250 m	500 m	1000 m	2500 m	5000 m

Tabela 5.3 - Relação do tamanho do cabo CAN com a taxa de transferência de dados

Assim, no software, a porta CAN1 é configurada para ter um *band rate* (taxa de transmissão de dados) de 1 Mb/s, já que o cabo que faz a conexão do terminal CAN1 do controlador de movimentos ao slot 3 do servo-drive tem tamanho menor que quatro metros. Essa configuração pode ser vista na figura 5.3.



Figura 5.3 - Configuração da taxa de transferência de dados

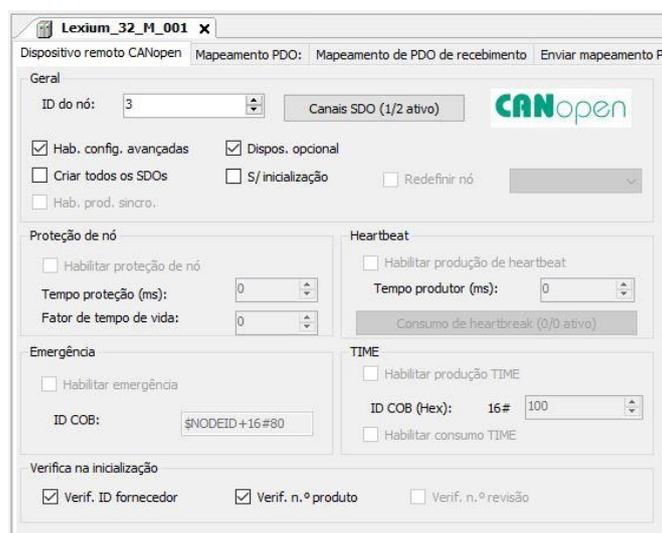


Figura 5.4 - Configuração do endereço do nó do servo-drive

À entrada CAN1 do software, são escolhidos então os três dispositivos que serão conectados, no caso o servo-drive LXM32MD18M2 e os dois servo-drives LXM32MU90M2 que são, então, configurados também para essa taxa de transmissão. Cada drive é configurado para um endereço de nó, sendo o eixo X configurado para o nó 1, o eixo Y para o nó 2 e o Z, por fim, para o nó 3, visto na figura 5.4. A configuração dos nós e da taxa de transmissão de dados devem ser realizadas tanto no software quanto diretamente nos drives, para que a comunicação do controlador de movimentos esteja corretamente pareada com os drives.

Após configurar o endereço dos nós de cada um dos servo-drives, é preciso configurar o mapeamento PDO (*Process Data Object* – Objeto de processamento de dados, ver capítulo 2). Para os servo-drives foi configurado o modo CSP (*Cyclic Synchronous Position*), como pode ser visto na figura 5.5. Neste modo, o controlador do motor recebe uma especificação posicional absoluta em intervalos de tempo fixos (chamados "ciclos") através do barramento de campo. A posição final é transferida via PDO, para a qual o controlador do motor responde prontamente.

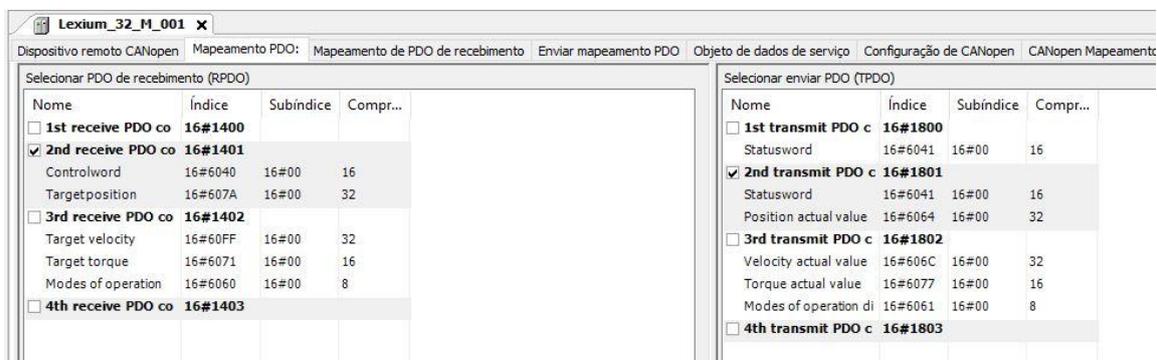


Figura 5.5 - Configuração do mapeamento PDO

Logo após, é feita a configuração da resolução do servo motor, como visto na figura 5.6. No manual dos servo-drives é possível ver que os servos possuem uma resolução de 131072 pontos/volta, o que significa que cada volta que o motor faz, ele percorre essa quantidade de pontos. Assim, como o software trabalha com números hexadecimais, é configurado uma taxa de escalonamento de 16#20000 incrementos para um giro do motor.

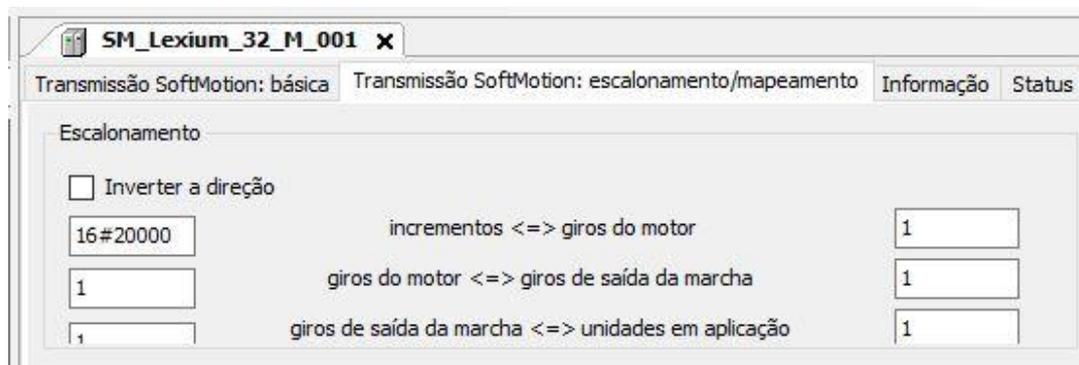


Figura 5.6 - Configuração do escalonamento do motor

Por fim, após todas as configurações realizadas via software, deve-se calcular o período de ciclo de transmissão de dados. Essa tarefa é de extrema importância na configuração para que não ocorra perda de transmissão de dados ou falha de sincronização. O cálculo desse tempo mínimo para troca de dados é mostrado na tabela 5.4.

Taxa de transferência de dados		250 Kbit/s	500 Kbit/s	1 Mbit/s
Modo CANmotion	Base	Tempo (ms)		
Tráfego padrão	Por conexão	1.01	0.66	0.48
Drive no modo CSP	Por drive	0.92	0.46	0.23

Tabela 5.4 - Tempo mínimo para transferência de dados para conexão CANopen

Assim, de acordo com a tabela o tempo mínimo necessário para que a transferência de dados ocorra sem perda de conexão é de 1,17 ms. Por razões de segurança, o valor realmente trabalhado deve ser pelo menos 20% maior que o resultado obtido. Por esse motivo, foi decidido trabalhar com um tempo de dois milissegundos, como pode ser visto na figura 5.6, no menu Período do ciclo (us).

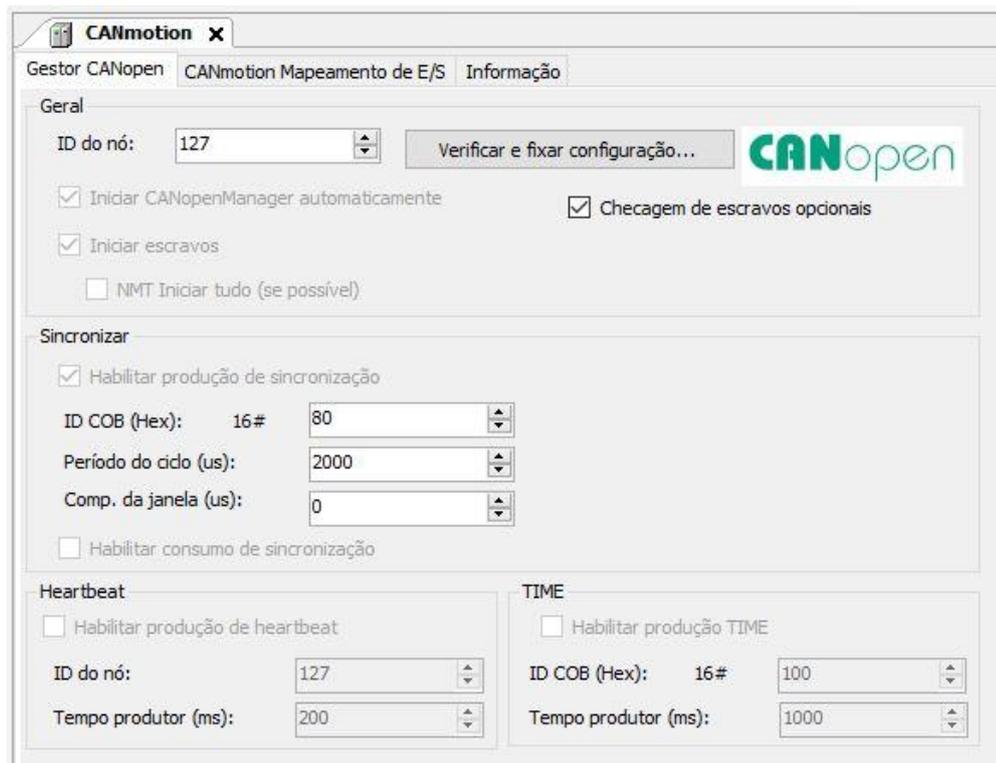


Figura 5.7 - Configuração do período de ciclo para transferência de dados

Além da configuração dos nós, é também configurado nos drives o método para realização da função *homing*. Essa função é responsável por encontrar os limites positivos ou negativos de cada eixo. Por esse motivo, os cabos do sensor de fim de curso foram conectados aos drives, sendo o limite negativo

conectado ao terminal DI3 e o positivo ao terminal DI2. Após a configuração desses terminais, é feita a configuração do *homing* para o método 1, que encontrará o limite negativo dos eixos e o torna o ponto referencial para a movimentação cartesiana, sendo este o ponto (0,0,0). A realização do *homing* pode ser feito tanto via software, por meio do bloco de função MC_Homing, quanto diretamente no drive. O bloco de função pode ser visto na figura 5.13.

5.2 Teste de locomoção dos eixos

Para a realização do teste de locomoção dos eixos, foi utilizado um programa-exemplo encontrado no próprio software. O programa utilizado se chama CANmotion_LXM, que faz a movimentação apenas de um eixo. Esse programa mostra que a movimentação do eixo do robô é feita a partir da quantidade de rotações que o servo-motor do eixo deve fazer. Sua interface principal pode ser vista na figura 5.8 Assim, se for requisitado que o robô vá para posição 60, por exemplo, significa que o motor deve fazer 60 rotações.

Para esse projeto, os pontos que o robô deve alcançar são dados em coordenadas cartesianas, em que cada ponto tem uma distância de 1mm entre si. Como para a programação do robô é utilizado a quantidade de voltas que o motor deve dar como entrada, quando é requisitado que o eixo se mova para um certo ponto em coordenada cartesiana, o valor desse ponto é multiplicado por um fator de correção, que será a entrada para a quantidade de rotações que motor deve realizar, para que o atuador seja deslocado para a posição que foi escolhida pelo operador da máquina.

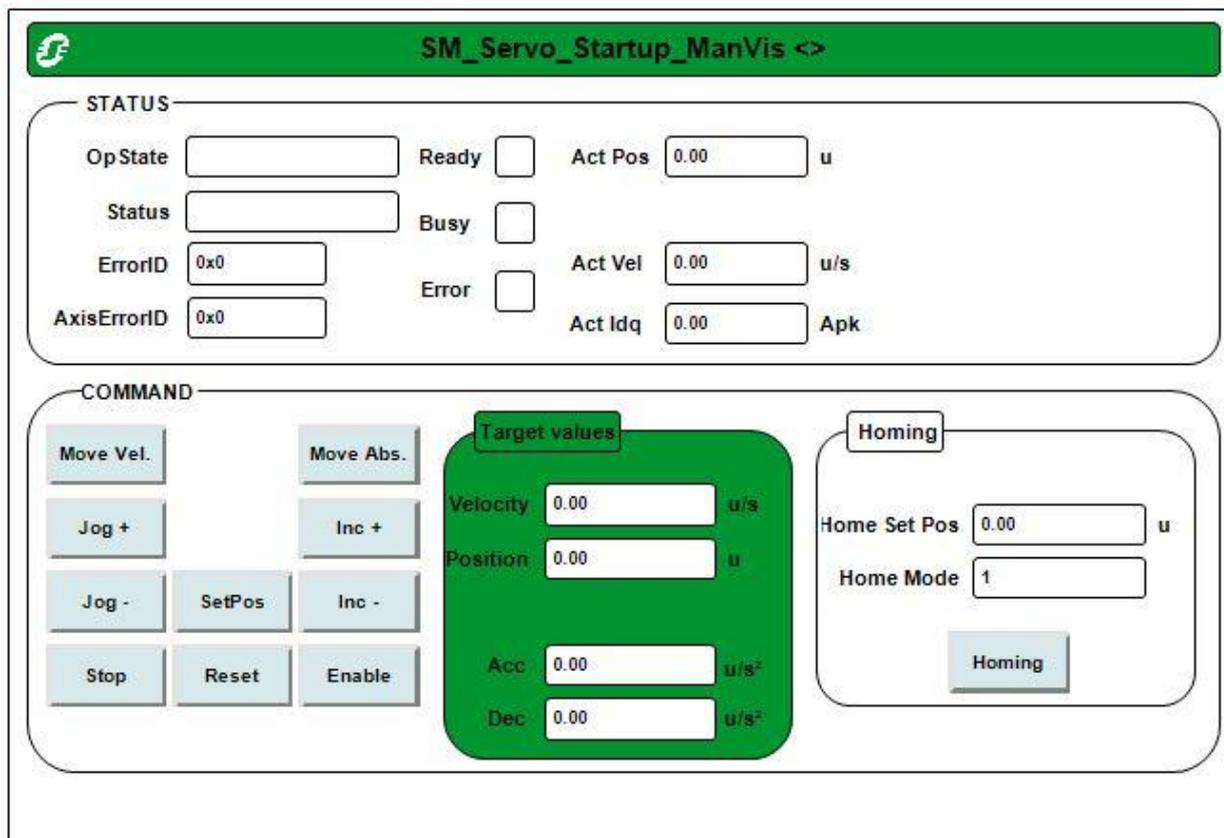


Figura 5.8 - Interface do programa utilizado para o teste de movimentação dos eixos

5.3 Algoritmo

A confecção do algoritmo para realização do programa foi feita de acordo com os passos mostrados no fluxograma da figura 5.9.

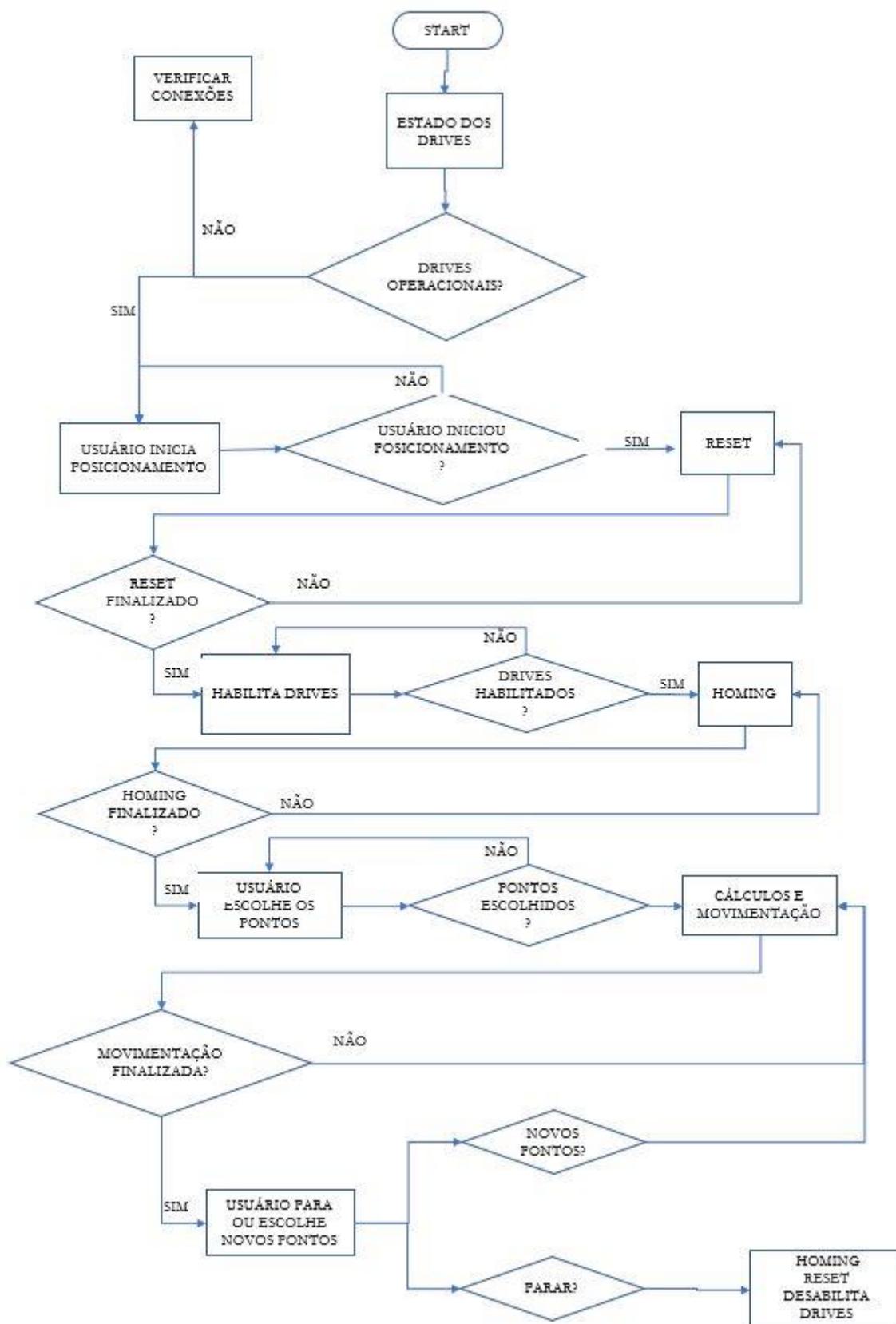


Figura 5.9 - Fluxograma do algoritmo

O primeiro passo é verificar se os drives estão operacionais. Se eles não estiverem, significa que há problema com a conexão ou configuração e o software não irá conectar corretamente com o controlador e com os drives. Caso eles estejam operacionais, segue-se com o passo dois.

Com os drives operacionais, o botão para inicializar a operação de movimentação dos drives poderá ser acionado. Se o botão for pressionado, segue-se com o passo três; se ele não for pressionado, deve-se esperar até que o usuário o pressione para inicializar a operação.

Depois que o usuário pressiona o botão para inicializar as operações, o programa realizará primeiramente o reset dos drives. Nesta operação, os erros encontrados nos drives resultantes de operações anteriores serão resetados. Quando a operação estiver finalizada, segue-se o passo quatro.

No quarto passo, os drives são habilitados, só então é habilitada a alimentação dos servo-motores. O resto do programa só estará funcional se este passo for realizado, pois os blocos de função não são habilitados se os drives não estiverem realmente habilitados. Verifica-se o status da operação e se estiver ligado, segue-se o passo cinco.

Quando os drives estiverem habilitados e operacionais, o programa automaticamente realiza a operação de *homing*. Então, todos os eixos são movidos ao ponto inicial. Quando esta operação estiver finalizada, segue-se o passo seis.

No sexto passo, é habilitado o botão para selecionar as posições, e o usuário poderá digitá-las na tela. Quando o usuário realiza essa tarefa, segue-se o passo sete.

Os valores digitados pelo usuário são salvos em variáveis temporárias. Esses valores são multiplicados pelo fator de correção. O resultado será então a entrada da posição alvo dos blocos de função que comandam a movimentação dos eixos. Os valores das posições iniciais e posições alvo são utilizados para calcular as constantes da equação de quinta ordem para o planejamento de trajetórias. Essas constantes são, então, utilizadas para calcular a velocidade com a qual o motor irá rotacionar. Um bloco de função que realiza um pulso por segundo foi utilizado para ativar a função para que uma velocidade nova seja calculada por segundo e enviada à entrada de velocidade do bloco de função que comanda os movimentos do robô. Os valores das acelerações e desacelerações dos três eixos foram ajustados para 5m/s^2 .

Depois que a movimentação dos eixos é realizada, o usuário poderá escolher novos pontos para a movimentação do eixo. Faz-se então novamente a realização do passo sete. Caso o usuário queira parar a movimentação dos eixos, segue-se o passo nove.

Caso o usuário deseje parar a operação, ele pressionará o botão de parada e então o robô fará, primeiramente, a operação *homing*, depois irá resetar quaisquer erros que ocorrerem depois que o processo de *homing* estiver finalizado e, então, desabilitará a energia dos drives.

5.4 – Programa

Para a confecção do programa foram utilizados blocos de funções prontas que o próprio software fornece, para ligar os drives, resetar, fazer o *homing*, realizar a movimentação e leitura da posição. Além disso foram utilizados blocos para realização de pulsos e detecção de cliques, para os botões.

Para a realização da função para resetar os erros dos blocos, foi utilizado o bloco de função MC_Reset, visto na figura 5.10. Este bloco de função é encontrado na biblioteca SM3_Basic do software.



Figura 5.10 Bloco de função MC_Reset

O bloco de função é responsável por resetar os erros ocasionados nos drives por programas anteriores. É a primeira função realizada quando o programa está em andamento e possui duas entradas e quatro saídas, sendo elas:

- Entradas:

Axis: Recebe o nome do eixo que será movimentado. O nome do eixo é dado no momento em que se escolhe o dispositivo a ser controlado;

Execute: Quando esta entrada é ativada, o bloco de função é ativado e realiza a função de resetar os erros dos drives;

- Saídas:

Done: Essa saída é ativada quando o comando de reset é finalizado;

Busy: A saída está ativa enquanto o bloco estiver realizando o reset de erros dos drives;

Error: Quando acontece algum erro ao tentar resetar o drive, essa saída é ativada;

ErrorID: Em caso de erro, essa saída mostra qual foi o erro ocorrido;

Quando o reset dos erros é finalizado, é, então, ativado o bloco de função que habilita o envio de energia do servo-drive ao servo motor. Para isso, é utilizado o bloco de função MC_Power, visto na figura 5.11. Esse bloco de função é também da biblioteca SM3_Basic.

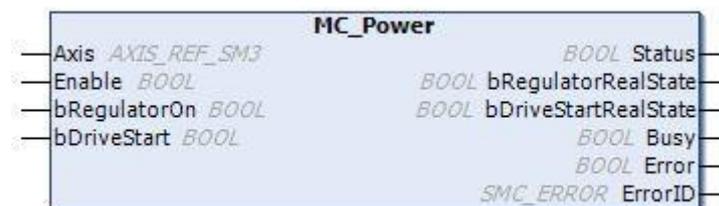


Figura 5.11 Bloco de função MC_Power

O bloco de função é dotado de quatro entradas e seis saídas, sendo elas:

- Entradas:

Axis: Recebe a variável eixo que será movimentado;

Enable: Quando esta entrada estiver ativa, o bloco de função é ativado;

bRegulatorOn: Entrada que habilita o envio de energia do drive ao motor;

bDriveStart: Quando esta entrada está ativa, o mecanismo de *Quickstop* (parada brusca, sem desaceleração) é desativado;

- Saídas:

Status: Quando todas as entradas estiverem habilitadas, esta saída estará verdadeira. Assim, ela mostrará que os drives estão prontos para realizar movimentações;

bRegulatorRealState: Mostra o estado efetivo do bloco de função, ou seja, se o bloco está realmente ativo. Caso esteja falso, o bloco não está habilitado;

bDriveRealState: Mostra se o mecanismo de *Quickstop* está realmente desativado;

Error: Mostra se ocorreu algum erro no processo de habilitação do bloco de função. Caso esteja ativada, o bloco de função não será ativado e a movimentação dos motores não será possível;

ErrorID: Identifica qual erro ocorreu no processo de habilitação do bloco de função;

Para que este bloco de função seja realmente habilitado, todas as entradas devem estar setadas.

Depois que o bloco de função MC_Power é habilitado e está ativado, o botão para iniciar a movimentação do robô e iniciar o teste é habilitado. Para isso, foi utilizado o bloco de função R_Trigger, visto na figura 5.12, que detecta cliques de botão para ativar variáveis. Esse bloco de função se encontra na biblioteca Standard.



Figura 5.12 - Bloco de função R_Trigger

Esse bloco de função possui apenas uma entrada e uma saída. A entrada CLK recebe o clique do botão, enquanto a saída Q mostra se o botão foi pressionado ativando, assim, as próximas linhas do código.

Quando o programa é ativado para iniciar as movimentações, primeiramente é realizado o comando *homing*, assim os eixos são mandados para os seus respectivos pontos iniciais, para que a movimentação seja inicializada do ponto (0,0,0). Para realizar essa tarefa, o bloco de função MC_Home, figura 5.13, é utilizado.



Figura 5.13 - Bloco de função MC_Home

Esse bloco de função é dotado de três entradas e cinco saídas, sendo elas:

- Entradas:
 - Axis: Referência ao eixo de movimento;
 - Execute: Quando ativado, o bloco inicia a função;
 - Position: Seta a posição. No caso, é utilizado o valor zero para que, quando o drive atinja a posição inicial, esta seja a posição zero;
- Saídas:
 - Done: Ativado quando o comando estiver finalizado;
 - Busy: Ativado enquanto o comando está em execução;
 - CommandAborted: Em caso de problema na execução do bloco, esta saída é ativada e o movimento é parado;
 - Error: Em caso de erro na movimentação, a saída é ativada;
 - ErrorID: Mostra o motivo para o erro ter ocorrido;

Quando o bloco de função estiver finalizado a tarefa, o usuário poderá escolher o ponto para o qual ele deseja que o robô se movimente. Quando o usuário escolher os pontos, ele clicará em um botão que ativará os processos de cálculo do fator de correção, das constantes da equação de quinta ordem do planejamento de trajetória e do valor da velocidade que os motores irão rotacionar para que os três eixos terminem o posicionamento ao mesmo tempo. O botão é ativado utilizando novamente um bloco de função R_Trig. Quando a saída desse bloco de função é ativada, é também ativado o bloco de função BLINK, que pode ser visto na figura 5.14, da biblioteca Util, que mandará um pulso por segundo, assim uma variável “tempo” irá variar seu valor.

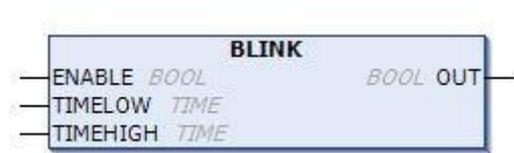


Figura 5.14 - Bloco de função BLINK

Esse bloco de função possui três entradas e uma saída. A entrada ENABLE habilita o funcionamento do bloco. Nas entradas TIMEHIGH e TIMELOW são colocados valores de tempo em milissegundos. Como o projeto prevê que o valor do pulso será por segundo, foram colocados 500ms nas duas entradas. Assim, o pulso será positivo durante os 500ms de TIMEHIGH e negativo durante os 500ms de TIMELOW.

No primeiro pulso, o valor da variável “tempo” será um, no segundo pulso será dois e no terceiro pulso será três. Assim, com esses valores serão calculados os valores das velocidades com as quais o eixo do motor irá se movimentar. Para realizar a movimentação dos eixos, foi utilizado o bloco de função MC_MoveAbsolute, que pode ser visto na figura 5.15, da biblioteca SM3_Basic.

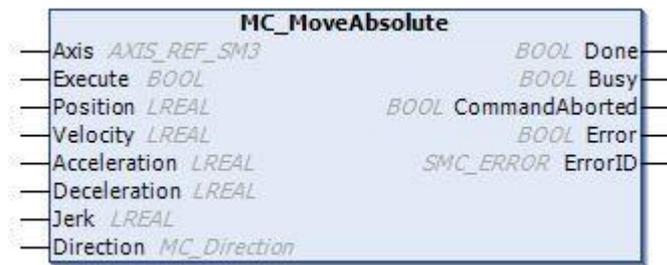


Figura 5.15 - Bloco de função MC_MoveAbsolute

Esse bloco de função possui oito entradas e cinco saídas, sendo elas:

- Entradas:

Axis: Referência ao eixo de movimento;

Execute: Quando setada, inicia a movimentação ao ponto escolhido;

Position: Valor para o qual o eixo deve se movimentar;

Velocity: Velocidade máxima com a qual o eixo do motor poderá alcançar, mas não necessariamente o alcança. Esse valor é calculado pela equação de quinta ordem do planejamento de trajetórias;

Acceleration: Valor da aceleração do motor, sempre positiva. Setada em 5m/s^2 ;

Deceleration: Valor da desaceleração do motor, sempre positiva. Setada em 5m/s^2 ;

Jerk: Valor do arranque, derivada da aceleração. Setada em 0m/s^3 ;

Direction: Direção que o eixo fará. Para eixos cartesianos pode ser positiva ou negativa;

- Saídas:

Done: Valor é verdadeiro quando a movimentação estiver finalizada;

Busy: Valor verdadeiro enquanto o eixo estiver realizando a movimentação;

CommandAborted: Verdadeiro quando, por algum motivo, a movimentação teve que ser parada. Por exemplo, quando ele bate em algum obstáculo ou o eixo encontra o sensor de fim de curso;

Error: Verdadeiro caso ocorra alguma falha na movimentação;

ErrorID: Mostra o motivo para o erro ter ocorrido;

A todo momento o valor da posição atual do eixo está sendo verificada. Seu valor é utilizado sempre que o usuário mandar os eixos do motor irem para alguma posição. Assim, a posição em que ele se encontra é utilizada como posição inicial para os cálculos das constantes da equação do planejamento de trajetória. Para esse fim, foi utilizado o bloco de função MC_ReadActualPosition, da biblioteca SM3_Basic. Esse bloco de função pode ser visto na figura 5.16.



Figura 5.16 - Bloco de função MC_ReadActualPosition

Esse bloco de função possui duas entradas, sendo elas Axis, referência ao eixo de movimento, e Enable, para habilitar o bloco de função. A saída Valid está setada caso o bloco de função esteja ativo, Busy fica ativo enquanto o bloco de função estiver lendo a posição do eixo. As saídas Error e ErrorID mostram se ocorreu algum erro e o motivo para o erro ter ocorrido, respectivamente, e a saída Position mostra a posição atual em que o eixo se encontra, de acordo com a leitura feita do *encoder* do motor.

O programa inteiro pode ser visto no Anexo IV.

CAPÍTULO 6 – Resultados

6.1 Instalação dos dispositivos no painel de controle

Ao ser ligado pela primeira vez, pôde ser percebido que o dispositivo redutor de tensão utilizado para alimentação dos *coolers* estava consumindo uma potência muito elevada, o que impossibilitaria a correta alimentação dos outros dispositivos que também eram conectados à fonte. Portanto, ele precisou ser retirado e foi concluído que os *coolers* deverão ser alimentados por uma fonte própria de 12V. Fora esse fato, a instalação dos dispositivos no painel ocorreu de forma satisfatória.

No Apêndice IV podem ser vistas as imagens do resultado da instalação do manipulador e dos dispositivos no painel de controle.

6.2 Programa-exemplo e teste de locomoção dos eixos

O servo-motor do eixo X, BMH0702P06A2A possui gearbox PLE80 com redução 8:1, enquanto os servo-motores do eixo Y, BMH0702P06A2A, e do eixo Z, BMH0701P06F2A, possuem gearbox PLE60 também com redução 8:1. Essa taxa de redução significa, justamente, que oito rotações do motor realizam uma rotação no eixo. A partir da realização deste teste pôde ser comprovado que cada rotação que o eixo realiza, faz o atuador se mover de uma distância de 155 mm, e que uma rotação do eixo corresponde a oito rotações do motor. Assim, foi calculado que uma rotação do motor corresponde a uma movimentação de 19,375 mm do atuador. Então, pode-se constatar que o valor escolhido para o fator de correção, de 0,05613, está de acordo para relacionar os valores em milímetros com a quantidade de rotações que o motor deve fazer para alcançar a posição desejada pelo usuário.

Os resultados obtidos nos testes de movimentação para aquisição das características de rotação do eixo se mostraram satisfatórias, já que mostrou estar de acordo com as especificações do servo-motor e da resolução de seu *gearbox*.

6.3 Programa final

6.3.1 Interface do usuário

Os testes para o programa desenvolvido foram realizados em etapas, para verificação dos resultados em cada um dos comandos enviados ao manipulador. Para inicializar o programa, o usuário deverá utilizar a tela de controle de movimentação, visto na figura 6.1. Essa tela permite que o usuário inicie o teste de posicionamento, pare e escolha o ponto desejado.

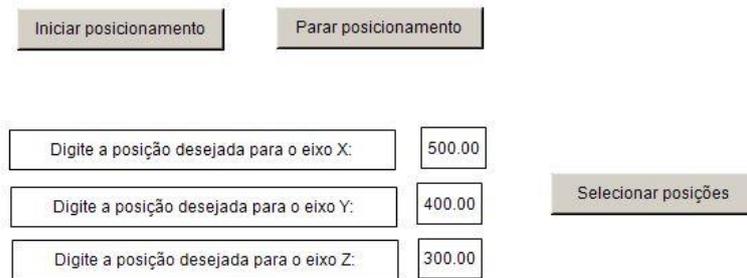


Figura 6.1 - Interface do usuário

Primeiramente o usuário deve clicar o botão “Iniciar posicionamento”. Quando esse botão é clicado, o programa irá inicializar o comando de reset dos drives. Quando essa operação estiver finalizada, acontece o comando *homing*, em que os eixos partem para as suas respectivas posições iniciais. Se essa operação for realizada com sucesso, o usuário escolhe, então, o ponto desejado e clica em selecionar posições. Caso o usuário deseje para o teste, basta clicar em “Parar posicionamento”. Os eixos realizam o comando homing novamente, e desliga o bloco de função MC_Power.

6.3.2 Teste de reset dos erros

Primeiramente foi feito o teste para verificar se o comando de reset dos drives estavam funcionando. Para essa verificação, foram utilizadas variáveis temporárias que mostram se o teste está em progresso e quando ele é finalizado. Em todos os testes realizados, foi possível confirmar que o comando foi realizado e finalizado com sucesso.

Quando a verificação do resultado do comando reset foi finalizada, foi realizada a verificação do comando para habilitar os drives e sua comunicação com os servo-motores. Foi também comprovado que o comando foi realizado com sucesso.

6.3.3 Teste de *homing*

O comando *homing* é realizado após a verificação do estado do bloco de função MC_Power. Quando a saída Status está verdadeira, é inicializada a operação *homing*. A realização deste comando se mostrou, por vezes, problemática.

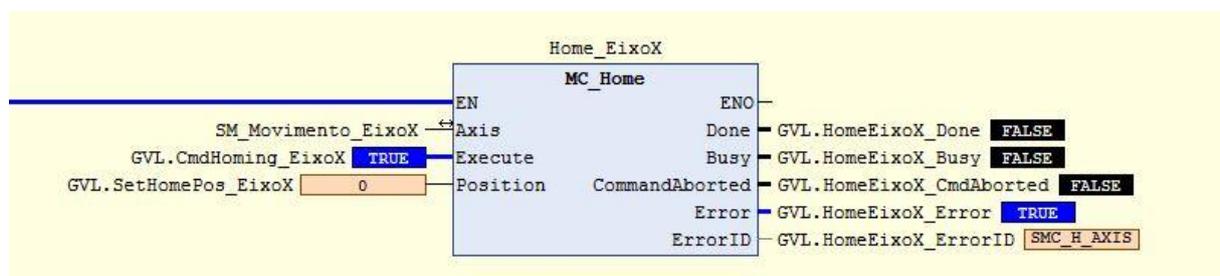


Figura 6.2 - Mensagem de erro após a realização do bloco de função MC_Home. Erro para o Eixo X.

Por exemplo, quando o atuador de algum eixo já se encontra na posição inicial, ocorre o erro A303. Este erro significa que o atuador está em contato com o sensor de fim de curso negativo e por esse motivo não poderá realizar nenhuma movimentação. A figura 6.2 mostra o resultado no bloco de função. Este erro ocorre normalmente nos eixos X e Y, e, por consequência deste erro nos eixos citados, ocorre o erro B121 no eixo Z. Esse erro ocorre quando dois sinais de comunicação são perdidos e, como diagnóstico é necessário verificar se a comunicação está ocorrendo de forma correta. Como se sabe que ocorreu erro no comando *homing*, constata-se que o erro ocorreu porque o sinal de comunicação entre os drives do eixo X e Y com o controlador de movimentos foi perdido. Para resolver este erro, é necessário que o programa seja excluído do controlador e, manualmente, o atuador com problema seja afastado do sensor de fim de curso.

Por vezes, mesmo realizando a operação de excluir o programa do controlador e afastar um pouco os atuadores do sensor de fim de curso, ocorre novamente o erro B121, desta vez nos três drives. O bloco MC_Home continua ativo, porém, como a conexão dos eixos foi perdida, o software para o programa imediatamente.

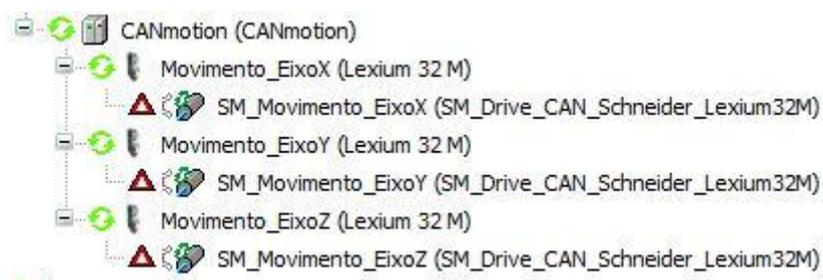


Figura 6.3 - Representação da perda de conexão

Quando ocorre perda de conexão, aparecem os triângulos vermelhos para demonstrar o ocorrido, como visto na figura 6.3. Para resolver esse problema é necessário excluir o programa do controlador e depois fazer novamente seu download. Porém a realização desta tarefa não é certeza de que não ocorrerá outro erro.

Por vezes, ocorre o erro B404, que significa perda de sinal de sincronia mais de duas vezes. Para resolver este problema, é necessário excluir o programa do controlador e verificar se a conexão do cabo CANopen está corretamente instalada.

Os erros na execução do bloco de função MC_Home costumam ocorrer principalmente em duas ocasiões: quando os atuadores estão muito afastados do ponto inicial de operação e quando os eixos já estão no ponto inicial de operação. As mensagens de erro são mostradas no HMI do próprio drive e na aba “Log” do software, que aparece quando o usuário seleciona o controlador.

Descrição
EXCEPTION Watchdog Application Motion
EXCEPTION ProcessorLoadWatchdog all applications all tasks
EXCEPTION Omitted cycle Watchdog Application MAST
Failed to resolve SysTaskEnd
Failed to resolve SysTaskResume
Failed to resolve SysTaskCreate
Application Applicationnot found to start
Bootproject Application corrupt. CRC File is not valid
Failed to resolve MemPoolDelete

Figura 6.4 - Mensagens de erro da aba Log

Quando as mensagens de erro da figura 6.4 aparecem, o programa para automaticamente e deve ser excluído do controlador. Porém, esses erros não representam nenhum risco ao controlador ou aos drives, podendo ser simplesmente ignorado.

Após algumas tentativas de utilização do programa, reinicializações do programa e desligamentos do painel de controle, o programa finalmente funciona. Quando ele realiza com sucesso o comando *homing*, o usuário está, então, apto a continuar o teste de posicionamento e escolherá o ponto para o qual o eixo deve se movimentar.

6.3.4 Teste de posicionamento dos eixos

Nesta operação, o usuário digita a coordenada para a qual o eixo deve se movimentar, como visto na figura 6.5, e clica no botão “Selecionar posições”. Ao fazer isso, o programa multiplica o valor digitado pelo fator de correção e envia esse valor para a entrada Position do bloco de função MC_MoveAbsolute. O valor da posição em que o robô se encontra no momento e o valor da posição para a qual os eixos devem se mover são utilizadas para calcular o valor das constantes da equação de quinta ordem do planejamento de trajetória. A partir dessas constantes, é então calculado o valor da velocidade com a qual o eixo irá se movimentar, de modo que os três eixos cheguem às suas respectivas coordenadas ao mesmo tempo, fazendo a movimentação em três segundos. Esse foi o tempo escolhido para a movimentação dos eixos quando feito o cálculo da equação de quinta ordem do planejamento de trajetórias.

Digite a posição desejada para o eixo X:	500.00
Digite a posição desejada para o eixo Y:	400.00
Digite a posição desejada para o eixo Z:	300.00

Figura 6.5 – Primeiro teste de posicionamento

No primeiro teste de posicionamento, foi requerido que os eixos se movessem ao ponto (500,400,300). Assim, esse valor foi multiplicado pelo fator de correção, como mostrado na figura 6.6.

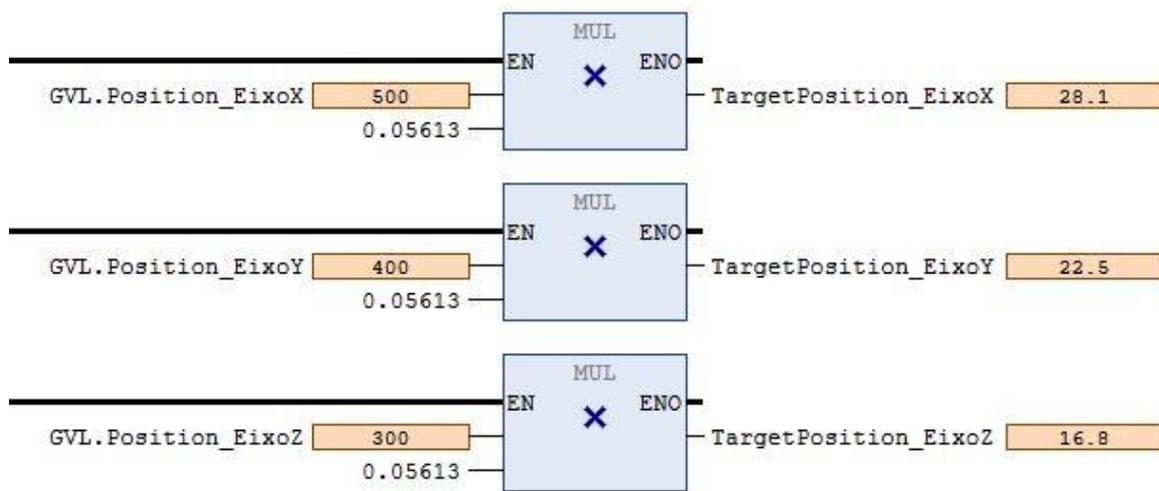


Figura 6.6 – Resultado da multiplicação pelo fator de correção

No primeiro teste, os eixos estavam nas suas posições iniciais, o que significa que os valores das variáveis TargetPosition representam a quantidade de rotações que o motor deve fazer.

Pelas figuras 6.7, 6.8 e 6.9 é possível verificar que os resultados obtidos para os comandos de movimentação dos eixos foram satisfatórios, visto que, pela leitura do resultado de saída do bloco de função MC_ReadActualPosition, responsável por ler o valor do encoder dos eixos, é possível notar que os atuadores se moveram para as coordenadas comandadas pelo usuário.

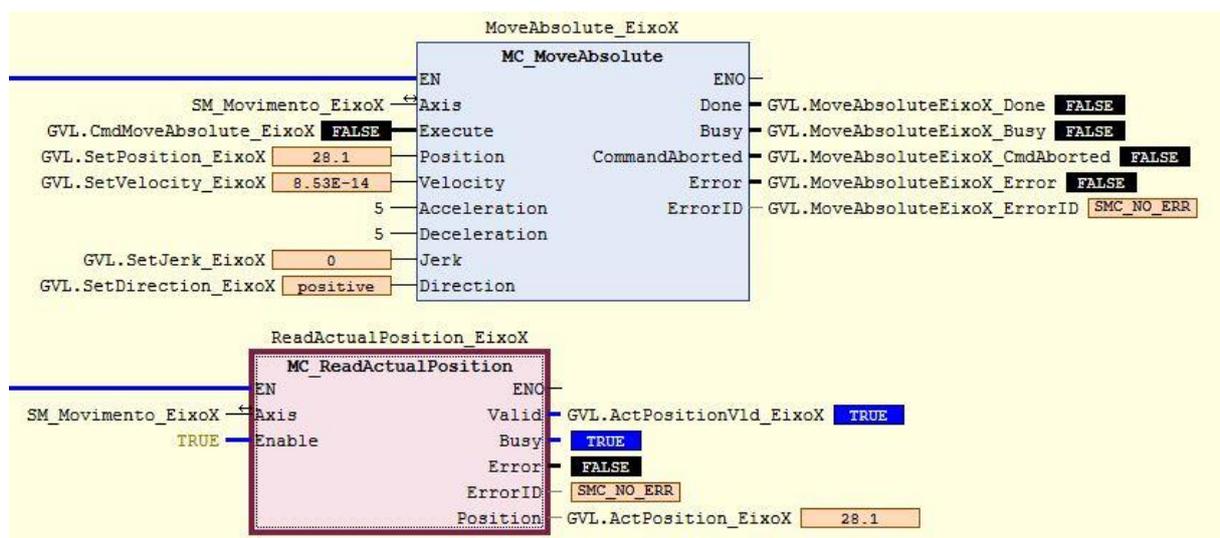


Figura 6.7 – Resultado do eixo X para o primeiro teste de movimentação

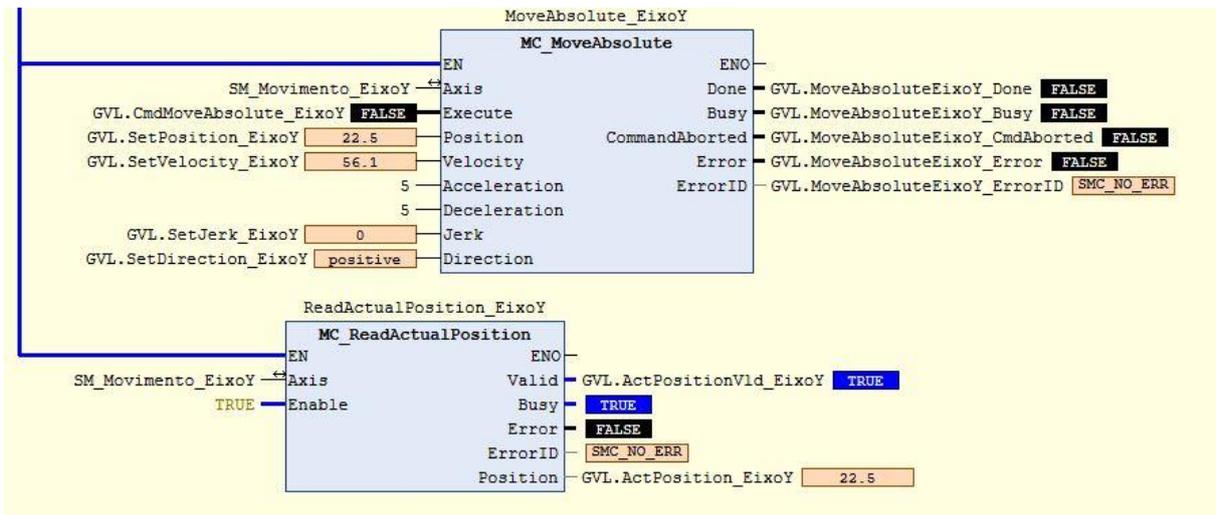


Figura 6.8 – Resultado do eixo Y para o primeiro teste de movimentação

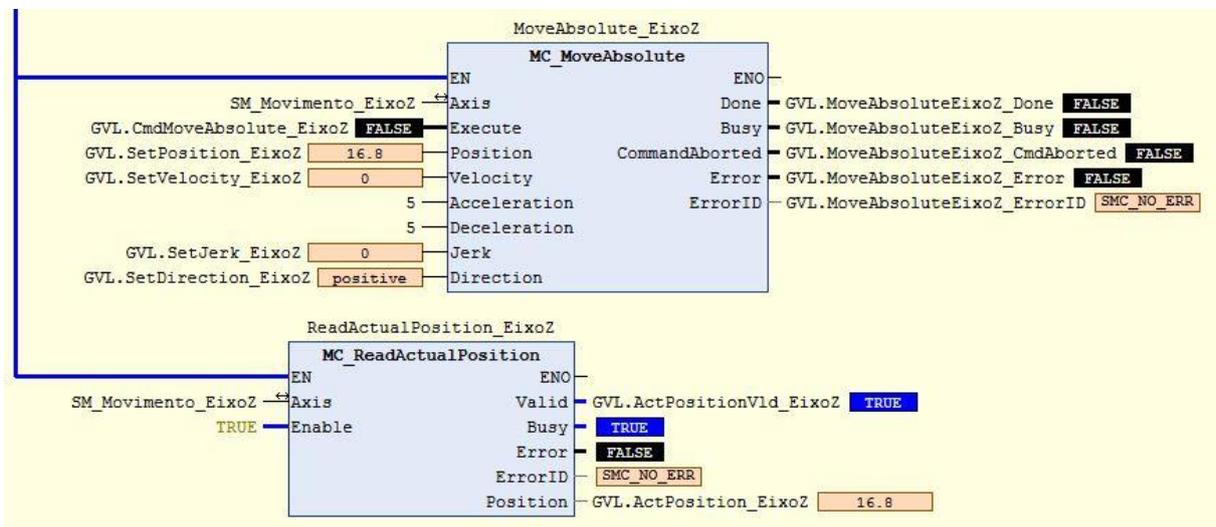


Figura 6.9 – Resultado do eixo Z para o primeiro teste de movimentação

No segundo teste de posicionamento, visto na figura 6.10, os eixos já estavam no ponto requerido anteriormente pelo usuário. O bloco de função MC_MoveAbsolute já é habilitado para calcular a quantidade de rotações que ele deve fazer para ir até certo ponto, não sendo necessário, assim, que seja calculada a quantidade de rotações que ele deve fazer para ir adiante ou voltar a alguma posição.

Digite a posição desejada para o eixo X:	700.00
Digite a posição desejada para o eixo Y:	300.00
Digite a posição desejada para o eixo Z:	200.00

Figura 6.10 – Segundo teste de posicionamento

Foi requerido, nesse teste, que o atuador do eixo X fosse adiante para a posição 700, enquanto os atuadores dos eixos Y e Z deveriam voltar às posições 300 e 200, respectivamente. Os resultados obtidos podem ser vistos nas figuras 6.11, 6.12 e 6.13.

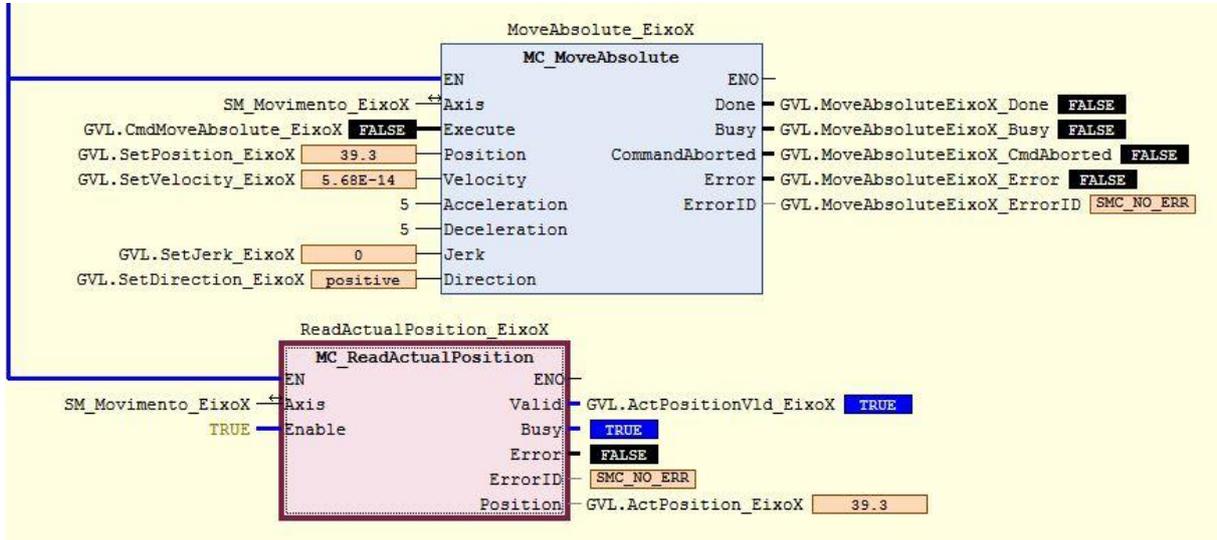


Figura 6.11 – Resultado do eixo X para o segundo teste de movimentação

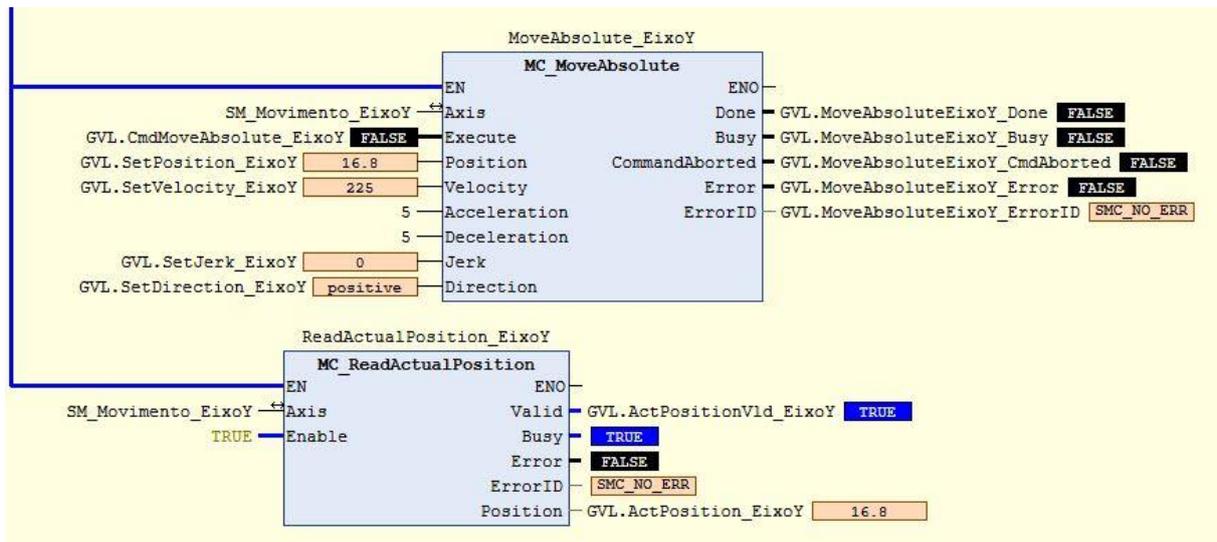


Figura 6.12 – Resultado do eixo Y para o segundo teste de movimentação

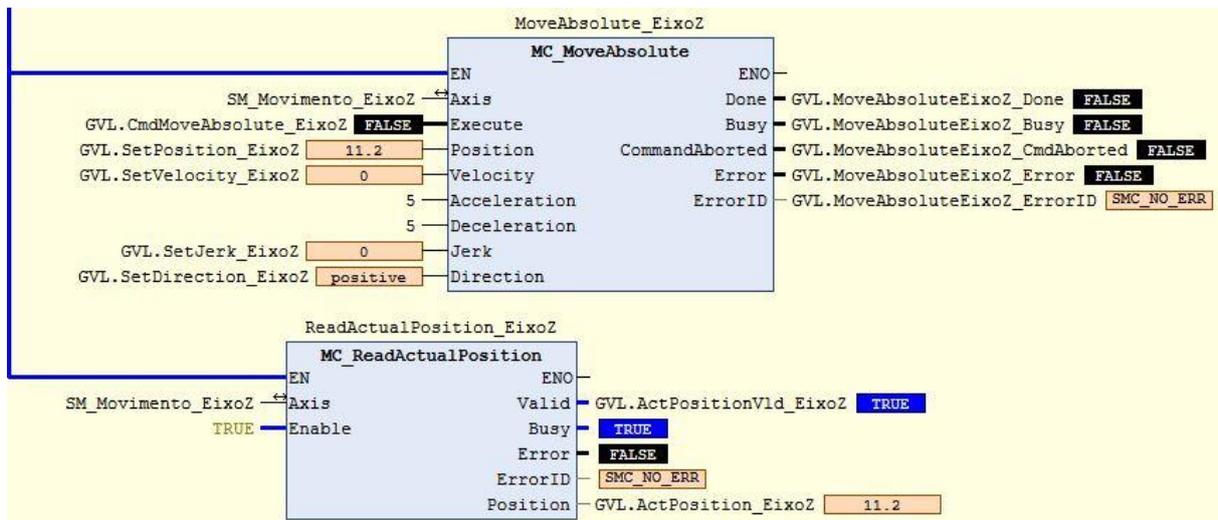


Figura 6.13 – Resultado do eixo Z para o segundo teste de movimentação

Os resultados para o segundo teste de movimentação se mostraram, também, satisfatórios, visto que os eixos se movimentaram exatamente ao ponto que foi desejado. Pela leitura dos resultados do bloco de função MC_ReadActualPosition pode ser notado que eles estão no ponto requerido.

Pode-se, então, verificar que os testes de utilização do bloco de função MC_MoveAbsolute se mostraram satisfatórios, notando, também, que sua utilização é precisa. O bloco faz a leitura da atual posição do eixo pelo encoder do drive e, sem necessidade de programação prévia, faz os atuadores se moverem de acordo com a posição que eles devem ir.

Após a realização deste teste, o usuário pode escolher parar a movimentação, bastando somente apertar o botão “Parar posicionamento”.

6.3.5 Teste do botão de parada

Quando o usuário desejar parar o teste, ele clicará no botão “Parar posicionamento”. Ao realizar esse comando, o programa inicia novamente o comando de reset, depois realiza o *homing* e, então, desabilita o bloco de função MC_Power.

Os testes mostraram que, novamente, o comando de reset funcionou de forma satisfatória. Entretanto, quando o comando *homing* está sendo realizado, os mesmos problemas citados na seção 4.3 podem ocorrer. Quando o comando *homing* é realizado com sucesso, o bloco de função MC_Power deve ser desabilitado. Para isso, as variáveis das entradas do bloco de função são resetadas, porém ele não realiza esse comando corretamente, pois suas saídas continuam verdadeiras, dando liberdade ao usuário de escolher novos pontos para os eixos se locomoverem. Em relação aos botões “Iniciar posicionamento” e “Parar posicionamento”, eles não estão mais habilitados.

CAPÍTULO 7 – Conclusão

O presente trabalho apresentou a instalação dos dispositivos de controle de um robô cartesiano, bem como o desenvolvimento de um programa que controla a movimentação dos seus eixos, com o intuito de aplicar os conhecimentos adquiridos durante o curso de engenharia mecatrônica e auxiliar o desenvolvimento de novos projetos a serem realizados para a melhoria do processo de automatização do grupo de automação e controle da universidade de Brasília. Esse trabalho deu continuidade ao projeto de graduação de um ex-aluno da universidade, cujo objetivo era escolher um robô a ser instalado no GRACO e selecionar os dispositivos de controle que fariam a movimentação de seus eixos.

Inicialmente, apresentou-se uma revisão bibliográfica sobre os assuntos que cercam o campo da robótica, em especial da robótica industrial e também foram apresentados os elementos que constituem um robô. Na revisão bibliográfica apresentou-se, ainda, o protocolo de comunicação CANopen, utilizado para fazer a conexão entre o controlador de movimentos e os drives que acionam os motores dos eixos robóticos, e técnicas de planejamento de trajetórias, mostrando os polinômios de terceira e quinta ordem.

A realização deste projeto de graduação foi dividida em quatro etapas. A primeira delas foi a pesquisa detalhada de trabalhos realizados na área de robótica industrial, utilizando como fonte de estudo livros, projetos de graduação anteriores, pesquisas feitas na área e, principalmente, os manuais dos dispositivos utilizados.

A segunda etapa contemplou o projeto em SolidWorks da distribuição dos dispositivos no painel de controle e o esquema de ligação elétrica destes dispositivos com o quadro geral de distribuição elétrica e com as fontes de alimentação.

Em seguida, com o projeto finalizado e com base nos estudos dos manuais dos dispositivos, deu-se início à terceira etapa, a instalação dos elementos no painel de controle. Esta foi a etapa mais desgastante deste trabalho, pois contemplou toda a alocação dos dispositivos no painel e todo o processo de fiação e alimentação. A terceira etapa foi de extrema importância para a obtenção do conhecimento sobre instalações elétricas e para demonstrar como esse processo é trabalhoso.

A quarta etapa, a programação utilizando o software SoMachine V4.1, foi, de fato, a mais demorada e complexa, tendo durado desde o início do projeto de graduação até a realização do relatório, ainda assim deixando muito o que aprender. A realização desta etapa foi especialmente complicada pela falta de projetos com os quais se basear e pela pouca bibliografia sobre o funcionamento do software e de seus blocos de função.

Os resultados obtidos das etapas três e quatro são interdependentes, visto que a programação só poderia ser realizada e testada no manipulador caso a instalação tivesse sido realizada com sucesso. Enquanto que somente seria possível verificar se a instalação fora feita corretamente, realizando-se os testes de programação. Assim, os resultados obtidos nas duas etapas foram satisfatórios, visto que a comunicação entre os dispositivos está ocorrendo de forma correta, ou seja, a comunicação entre o computador, o controlador de movimentos, os servo-drives e os servo-motores está ocorrendo conforme o desejado. Além disso, os resultados obtidos do programa mostram que o manipulador alcança as posições requeridas pelo usuário.

CAPÍTULO 8 – Considerações Finais

Em suma, é possível constatar que a execução deste trabalho foi de excepcional importância para a aplicação dos conhecimentos adquiridos no decorrer do curso. Em especial, as matérias Robótica Industrial e Elementos de Automação se mostraram singularmente importantes para quem deseja seguir carreira na área de robótica e realiza projetos de pesquisa nesta área. A efetuação deste projeto foi, em grande parte, baseada nos conhecimentos adquiridos pelo estudo dessas duas matérias.

Entretanto, apesar do vasto estudo e pesquisa realizados durante o processo de produção do presente trabalho, algumas falhas foram cometidas. A análise de posições dos eixos utilizando a representação de D-H não foi realizada. Assim, quando o usuário requisita que os eixos do robô se locomovam para um certo ponto, este ponto não está precisamente de acordo com as reais coordenadas cartesianas. Para resolver esse problema, seria necessário que mais testes fossem realizados a fim de se descobrir o real volume de trabalho do robô. Além disso, as mensagens de erro mostradas na seção 6.3.3 sugere que há problemas com a conexão entre o controlador de movimentos e os drives. Estes problemas podem estar ocorrendo devido a interferências ocasionadas pela instalação elétrica, má condição com cabo CANopen ou incorreta configuração dos drives, porém apenas testes podem dizer precisamente o motivo desses erros. Também não foi possível fazer a integração do controlador com o dispositivo HMI.

Todos esses problemas podem ser sanados em projetos futuros. A integração do dispositivo HMI com o controlador poderá facilitar o manuseio do robô e melhorar sua utilização. Também será necessário corrigir o problema que ocorre com a conexão entre o LMC058 e os servo-drives, já que a perda de conexão faz os atuadores pararem imediatamente de funcionar e pode causar problemas ainda maiores, como a quebra do próprio manipulador. Além disso, é necessário fazer a correta realização da análise de posições, para que, quando seja ordenado que o manipulador vá para uma posição, ele faça o percurso corretamente e alcance a posição real nas coordenadas cartesianas.

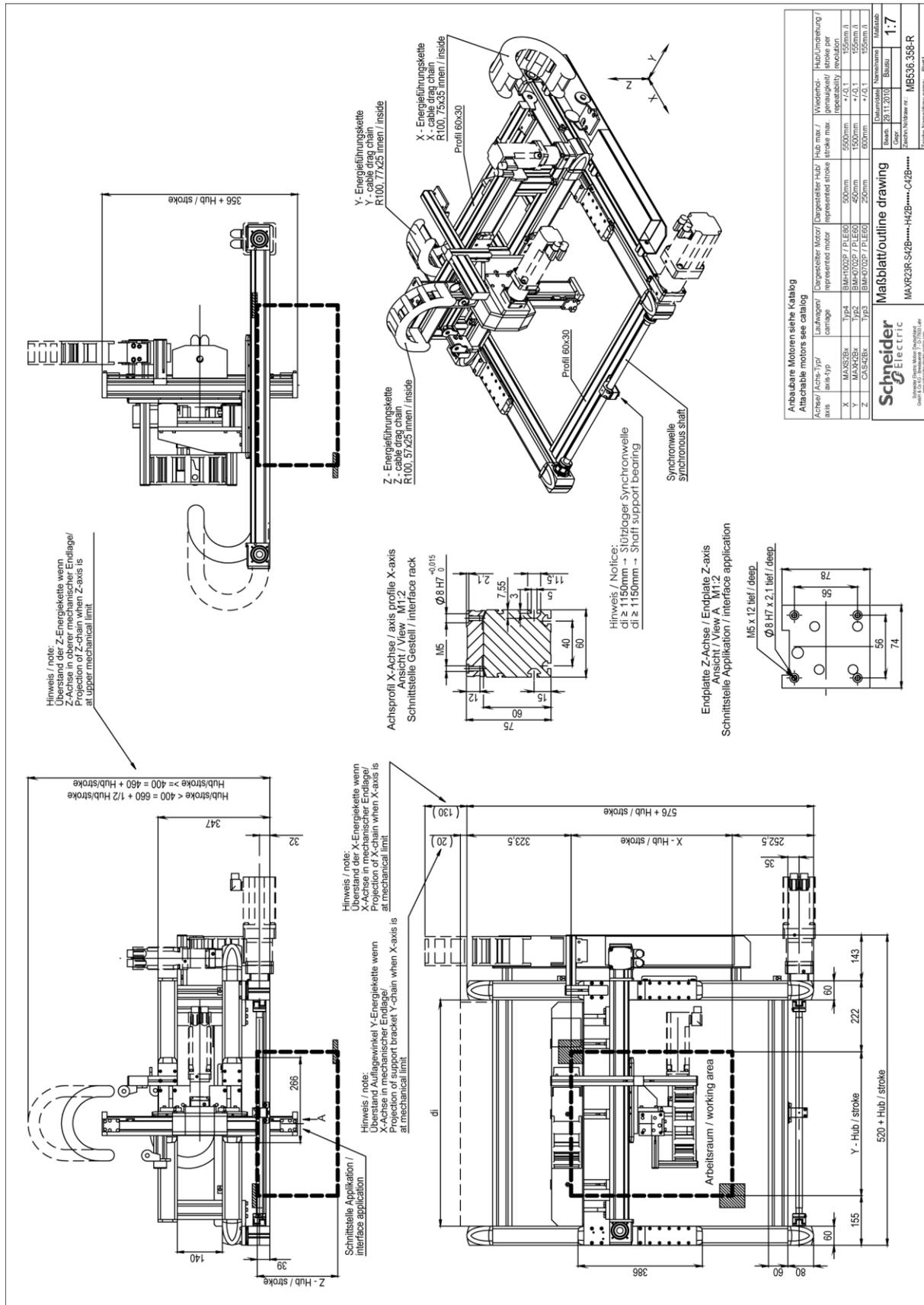
Além da resolução de problemas, em projetos futuros, este manipulador poderá ser também programado para processos automatizados de soldagem. A acoplagem de uma tocha de solda em seu efetuador terminal e a programação correta para os atuadores seguirem uma trajetória definida pode auxiliar o processo de soldagem de materiais além de melhorar os estudos na área. Além disso, câmeras podem ser acopladas ao robô para que o processo de soldagem seja inteiramente automatizado. A utilização das câmeras e o estudo detalhado da área de visão computacional podem auxiliar no desenvolvimento de um projeto que faça a automatização total do processo de soldagem, em que a câmera irá buscar a área e fazer a trajetória que deve receber a solda.

BIBLIOGRAFIA

- [1] Carrara, V. (s.d.). Apostilia de Robótica. Universidade Braz Cubas.
- [2] Hefferman, D., & Leen, G. (2002). A time-triggered control network for industrial automation. *Assembly Automation*, 60-68.
- [3] Heyer, C. (2010). Human-Robot Interaction and Future Industrial Robotics Applications. *The 2010 IEEE/RSJ International Conference on Intelligent Robots Systems*, (pp. 18-22). Taipei, Taiwan.
- [4] Pfeiffer, O., Ayre, A., & Keydel, C. (2008). *Embedded Networking with CAN and CANopen*. San Clement, CA: RTC Books.
- [5] Robertz, S. G., Henriksson, R., Nilsson, K., Blomdell, A., & Tarasov, I. (26-28 de Setembro de 2007). Using Real-time Java for Industrial Robot Control. Vienna, Austria.
- [6] Silva, T. M., Spohm, M. A., & Padilha, A. S. (2017). Uma análise comparativa entre os protocolos CANopen, DeviceNet e Smart Distributed System. *Revista Brasileira de Computação Aplicada*, 2-14.
- [7] Spong, M. W., Hutchinson, S., & Vidyasagar, M. (s.d.). *Robot Modeling And Control*. John Wiley & Sons, INC.
- [8] SCHNEIDER ELECTRIC. BMH Servo motor, Motor Manual, março, 2016. ISSN: 0198441113749 Disponível em: <<http://www.schneider-electric.com/manuals>> Acesso em 10 de março de 2018.
- [9] SCHNEIDER ELECTRIC. SoMachine Example Guide: CANmotion with a Lexium Drive. CANmotion_LXM.project, 01 de outubro de 2013. ISSN: EIO0000000926.01. Disponível em: <<http://www.schneider-electric.com>> Último acesso em julho de 2018.
- [10] CoDeSys Automation Alliance. Manual da biblioteca CAA_CiA405.lib. Disponível no software SoMachine V4.1.
- [11] SCHNEIDER ELECTRIC. Getting Started With SoMachine Self Study Manual, março, 2014. Disponível em <<http://www.schneider-electric.com>>.
- [12] SCHNEIDER ELECTRIC. Modicon LMC058, Motion Controller, Programming Guide, setembro de 2016. ISSN: EIO0000000408.08. Disponível em: <<http://www.schneider-electric.com>>. Acesso em abril, 2017.
- [13] SCHNEIDER ELECTRIC. SoMachine, FileFormatUtility, Library Guide, junho, 2017. ISSN: EIO0000002530.00. Disponível em: <<http://www.schneider-electric.com>>. Acesso em maio, 2018.
- [14] SCHNEIDER ELECTRIC. Lexium 32M, Servo Drive, User Guide, novembro, 2017. ISSN: 0198441113767.10. Disponível em: <<http://www.schneider-electric.com>> Acesso em março, 2018.
- [15] SCHNEIDER ELECTRIC. Modicon LMC058, Motion Controller, Hardware Guide, abril de 2017. ISSN: EIO0000000438.07. Disponível em: <<http://www.schneider-electric.com>>. Acesso em abril, 2018.
- [16] TODD, D. J. Fundamentals of robot technology: an introduction to industrial robots, teleoperators and robot vehicles. Londres: Kogan Page, 1986;
- [17] NIKU, S. B. Sérgio Gilberto Taboada Introdução à robótica: Análise, Controle, Aplicações. 2 ed. Rio de Janeiro: LTC, 2015;
- [18] Lamb, F. Márcio José da Cunha, Automação industrial na prática: Eixo, Controle e Processos Industriais. Porto Alegre: AMGH: 2015;

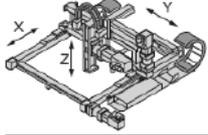
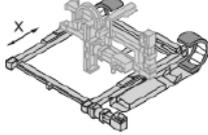
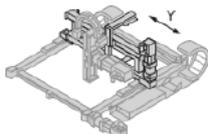
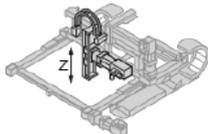
- [19] NEWSROOM. Quelle responsabilité pour un robot lors d'un accident? Disponível em: <<https://humanoides.fr/quelle-responsabilite-pour-un-robot-lors-dun-accident/>> Acesso em 15/12/2016;
- [20] Site do Aliexpress. Disponível em: <<https://pt.aliexpress.com/item/XYZ-three-axis-mechanical-three-dimensional-electric-slide-length-linear-module-customized-fast-delivery-manipulator/32379469512.html>> Acesso em: 15/12/2016;
- [21] Site da ABM Industrial Mining. Disponível em: <<http://www.abm-industrial.com/2013/02/07/que-es-un-encoder/>> Acesso em: 17/07/2017;
- [22] Site da Profelectro. Disponível em: <<http://www.profelectro.info/robotica-apontamentos-parte-8-ficha-de-trabalho-8-com-questionario-orgaos-terminais-e-sensores/>> Acesso em: 17/07/2017;
- [23] Site da Soluções Industriais. Disponível em: <<https://www.solucoesindustriais.com.br/empresa/maquinas-e-equipamentos/azetek/produtos/acessorios/atuadores-lineares>> Acesso em: 17/07/2017;
- [24] Site do Programa de Pós-Graduação em Energia da Universidade Federal do ABC Paulista. Disponível em: <<http://pgene.ufabc.edu.br/docentes/Riascos/ensino/disciplinas/Robotica/FundamentosRobotica.html>> Acesso em: 17/07/2017;
- [25] FERNANDES, M.O., (2016). Projeto de instalação de um robô cartesiano de 3 graus de liberdade. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-n°06, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 85p.
- [26] SCHNEIDER ELECTRIC. Lexium Linear Motion, Linear axes and multi-axis systems, Catalogue, abril de 2011. Disponível em: <<http://www.schneider-electric.com>>. Acesso em julho de 2017;
- [27] SCHNEIDER ELECTRIC. Magelis HMISTU655/855, User Manual. Disponível em: <<http://www.schneider-electric.com>>. Acesso em julho de 2017;
- [28] Associação Brasileira de Normas Técnicas. Norma Brasileira ABNT NBR 5410, 2004, Instalações elétricas de baixa tensão. Disponível em: <https://www.iar.unicamp.br/lab/luz/ld/normas%20e%20relat%F3rios/NRs/nbr_5410.pdf>. Acesso em julho de 2017;
- [29] SCHNEIDER ELECTRIC. Compact/CANmotion/Motion Controller/ LMC058 + Performance Packaging, System User Guide, Fevereiro de 2011. ISSN: EIO000000294.01. Disponível em: <<http://www.schneider-electric.com>>;
- [30] Site Rede de conhecimento computador. O que é um servo-drive. Disponível em: <<http://ptcomputador.com/Ferragens/computer-drives-storage/46896.html>> Acesso em 17/07/2017

Anexo I: Desenho técnico do robô



Anexo I.1 - Desenho técnico do MAXR23-S42-H42-C42 [26]

Anexo II: Característica dos atuadores

References (1)	
To order a MAX R●3 portal robot, complete each reference by replacing the "●" :	
Example: MAX R 13 R – S41 B R 3000 – H41 B R 1200 – C31 B C 1200 MAX R ● 3 ● – ●●● B ● ●●● – ●●● B ● ●●●	
+ PLE60 3:1 + BMH 0702P01A2A (for the X axis) + ...	
+ PLE60 3:1 + BMH 0702P01A2A (for the Y axis) + ...	
+ PLE60 3:1 + BMH 0702P01A2A (for the Z axis) + ...	
Size of X axis	40 (40 x 40 mm cross-section) 1
(profile cross-section)	60 (60 x 60 mm cross-section) 2
	80 (80 x 80 mm cross-section) 3
	110 (110 x 110 mm cross-section) 4
	
Number of axes	3 independent axes: 1 X axis, 1 Y axis, 1 Z axis 3
Interface for the drive element (3)	Drive element fixed on right-hand side R
	Drive element fixed on left-hand side L
Type of X axis	MAX S41 (for MAX R13) (2) S41
	MAX S42 (for MAX R23) (2) S42
	MAX S43 (for MAX R33) (2) S43
	MAX S44 (for MAX R43) (2) S44
	
Type of drive	Toothed belt B
Type of guide	Roller (for MAX R●3● – S41/S42/S43) R
	Ball (for MAX R●3● – S42/S43/S44) B
Stroke	State the length in mm (see the maximum possible length depending on the model ●●●● on page 61531/3)
Type of Y axis	MAX H41 (for MAX R13) (2) H41
	MAX H42 (for MAX R23) (2) H42
	MAX H43 (for MAX R33) (2) H43
	MAX H44 (for MAX R43) (2) H44
	
Type of drive	Toothed belt B
Type of guide	Roller (for MAX R●3● – S4●B●●●● – H41/H42/H43) R
	Ball (for MAX R●3● – S4●B●●●● – H42/H43/H44) B
Stroke	State the length in mm (see the maximum possible length depending on the model on page 61531/3) ●●●●
Type of Z axis	CAS 41 (for MAX R13) (2) C41
	CAS 42 (for MAX R23) (2) C42
	CAS 43 (for MAX R33) (2) C43
	CAS 44 (for MAX R43) (2) C44
	CAS 31 (for MAX R13) (2) C31
	CAS 32 (for MAX R23) (2) C32
	
Type of drive	Toothed belt B
Type of guide	Roller (for MAX R●3● – S4●B●●●● – H4●B●●●● – C41/C42/C43) R
	Ball (for MAX R●3● – S4●B●●●● – H4●B●●●● – C42/C43/C44) B
	Ball (for MAX R●3● – S4●B●●●● – H4●B●●●● – C3●) C
Stroke	State the length in mm (see the maximum possible length depending on the model on page 61531/3) ●●●●
Planetary gearbox gear ratio + motor reference	State the planetary gearbox gear ratio and the complete motor reference at the end of the reference, in plain text, selected for the X axis, the Y axis and the Z axis. Example: PLE60 3:1 + BMH 0702P01A2A for each axis + ...

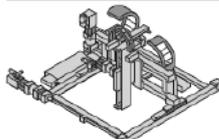
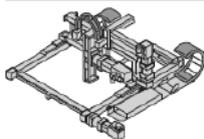
(1) All technical data (characteristics, dimensions, etc.) for Lexium MAX R●3 portal robots is available on our website www.schneider-electric.com and on the documentation CD-ROM supplied with this catalogue.

(2) Each axis is supplied with 2 PNP output sensors, NC contact, with a 0.1 m cable equipped with an M8 connector.

(3) Types of interface for the drive element:

MAX R●3R – ...

MAX R●3L – ...



Anexo II.1 - Característica dos atuadores [25]

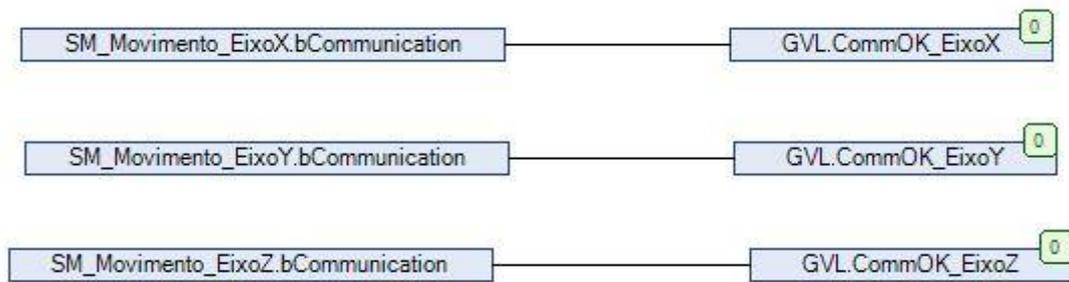
Anexo III: Configuração dos servo-motores

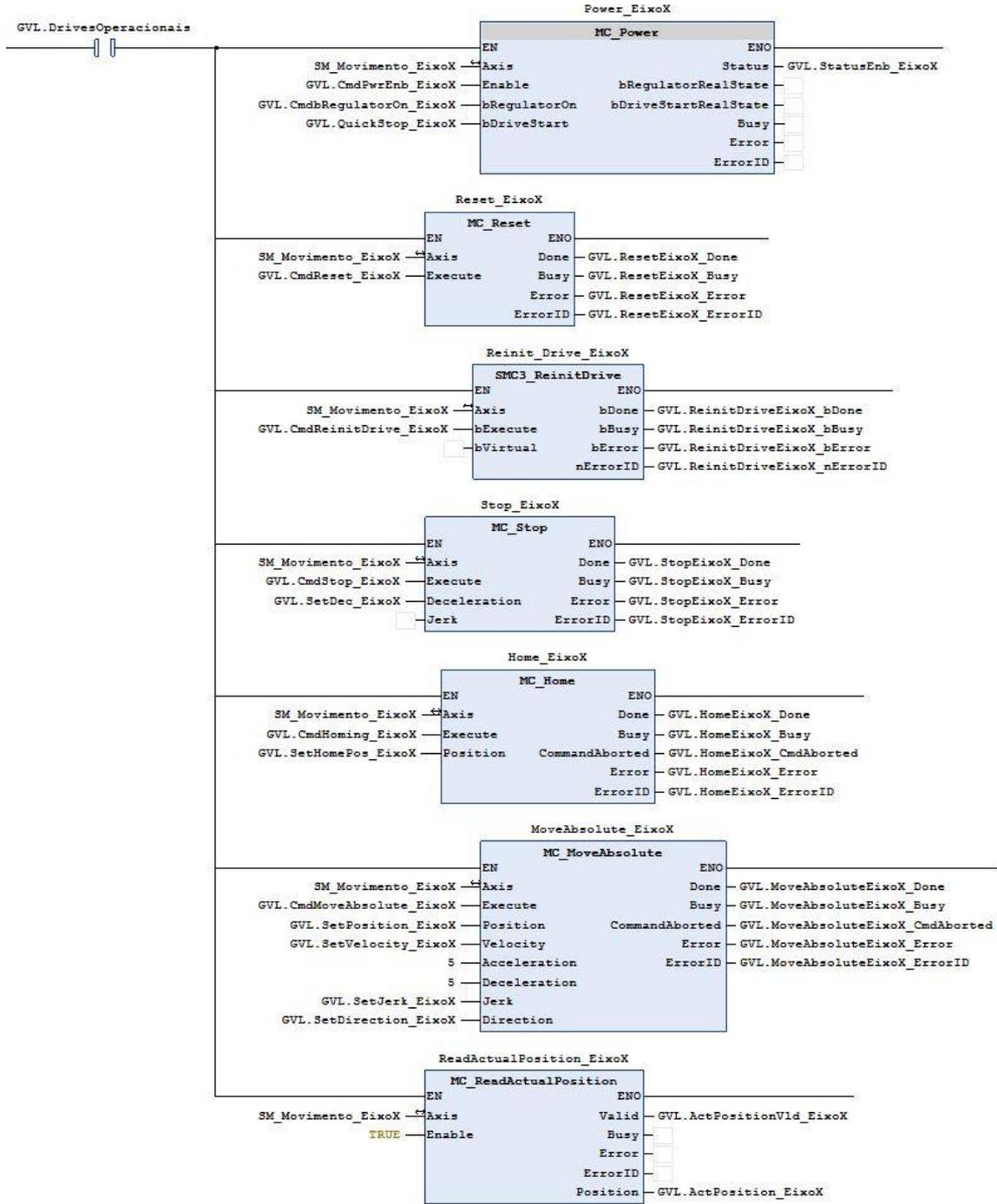
	BMH	070	1	P	0	1	A	1	A
Product family BMH: Synchronous motor - medium moment of inertia									
Size (housing) 070 = 70 mm flange 100 = 100 mm flange 140 = 140 mm flange 190 = 190 mm flange 205 = 205 mm flange									
Length 1 = 1 stack 2 = 2 stacks 3 = 3 stacks 4 = 4 stacks									
Winding P = Optimized in terms of torque and speed of rotation T = Optimized in terms of high speed of rotation									
Shaft and degree of protection 0 = Smooth shaft; degree of protection: shaft IP54 ¹⁾ , housing IP65 1 = Parallel key; degree of protection: shaft IP 54 ¹⁾ , housing IP 65 2 = Smooth shaft; degree of protection: shaft and housing IP65 ^{1) 2)} 3 = Parallel key; degree of protection: shaft and housing IP 65 ^{1) 2)}									
Encoder system 1 = Absolute singletum 128 Sin/Cos periods per revolution (SKS36) 2 = Absolute multitem 128 Sin/Cos periods per revolution (SKM36) 6 = Absolute singletum 16 Sin/Cos periods per revolution (SEK37) 7 = Absolute multitem 16 Sin/Cos periods per revolution (SEL37)									
Holding brake A = Without holding brake F = With holding brake									
Connection version 1 = Straight connector 2 = Angular connector 90°, can be rotated									
Mechanical interface - mounting A = International IEC Standard B = International IEC standard and fan cooling									

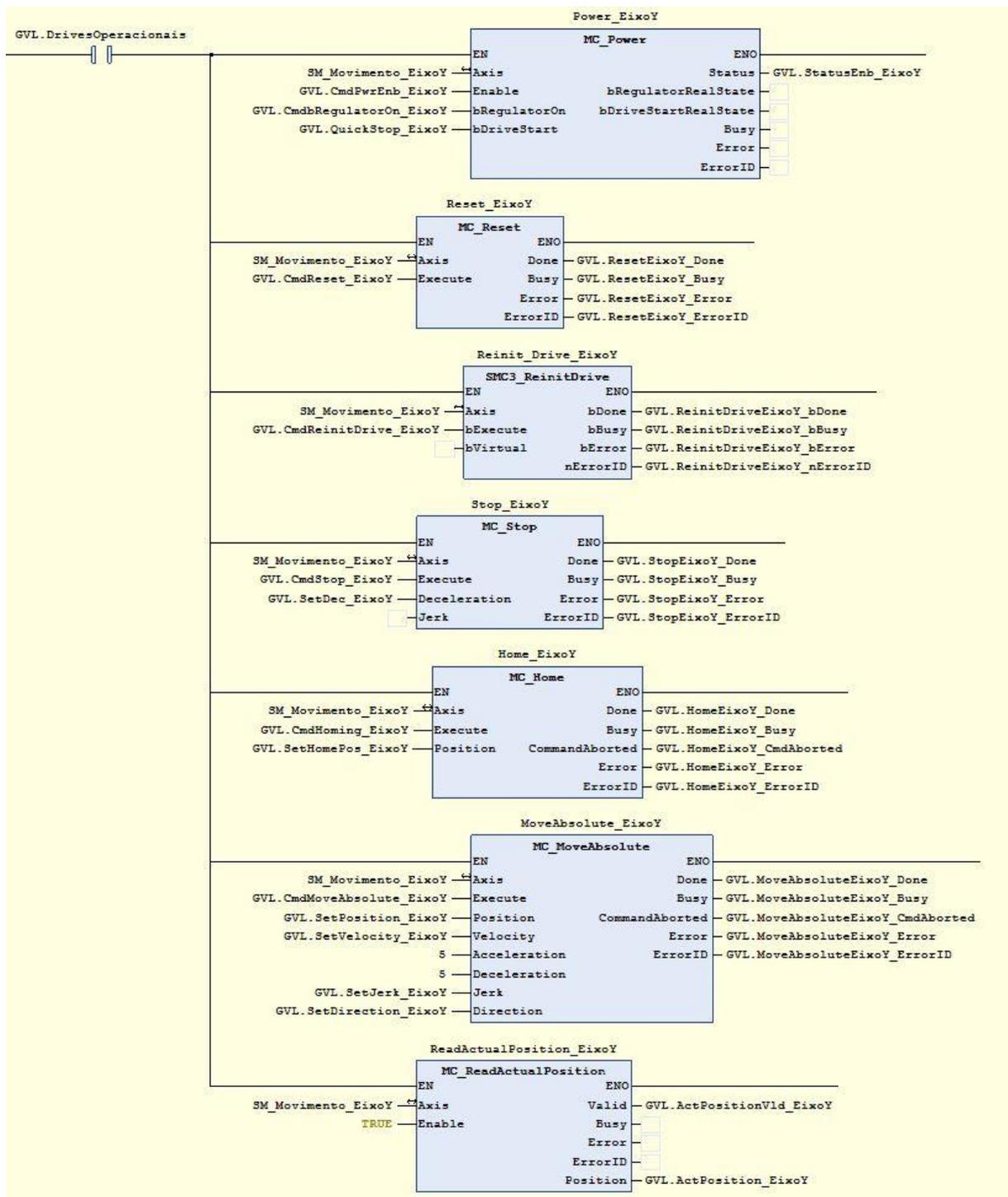
- 1) In the case of mounting position IM V3 (drive shaft vertical, shaft end up), the motor only has degree of protection IP50.
- 2) The maximum permissible speed of rotation is limited to 6000 rpm by the shaft sealing ring. Separate accessories allow you to obtain degree of protection IP67. See chapter "6 Accessories and spare parts".

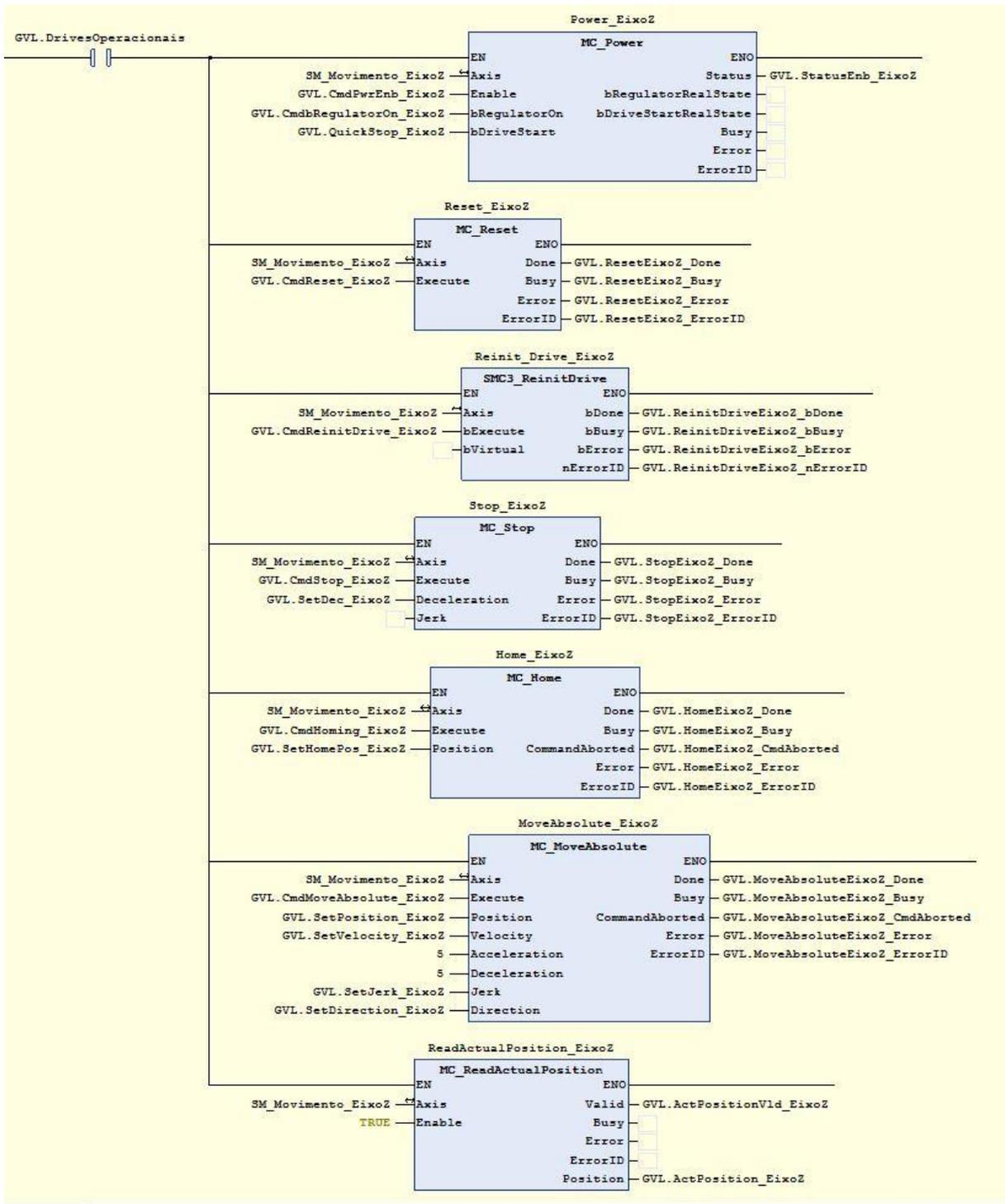
Anexo III.1 - Configuração dos servo-motores [25]

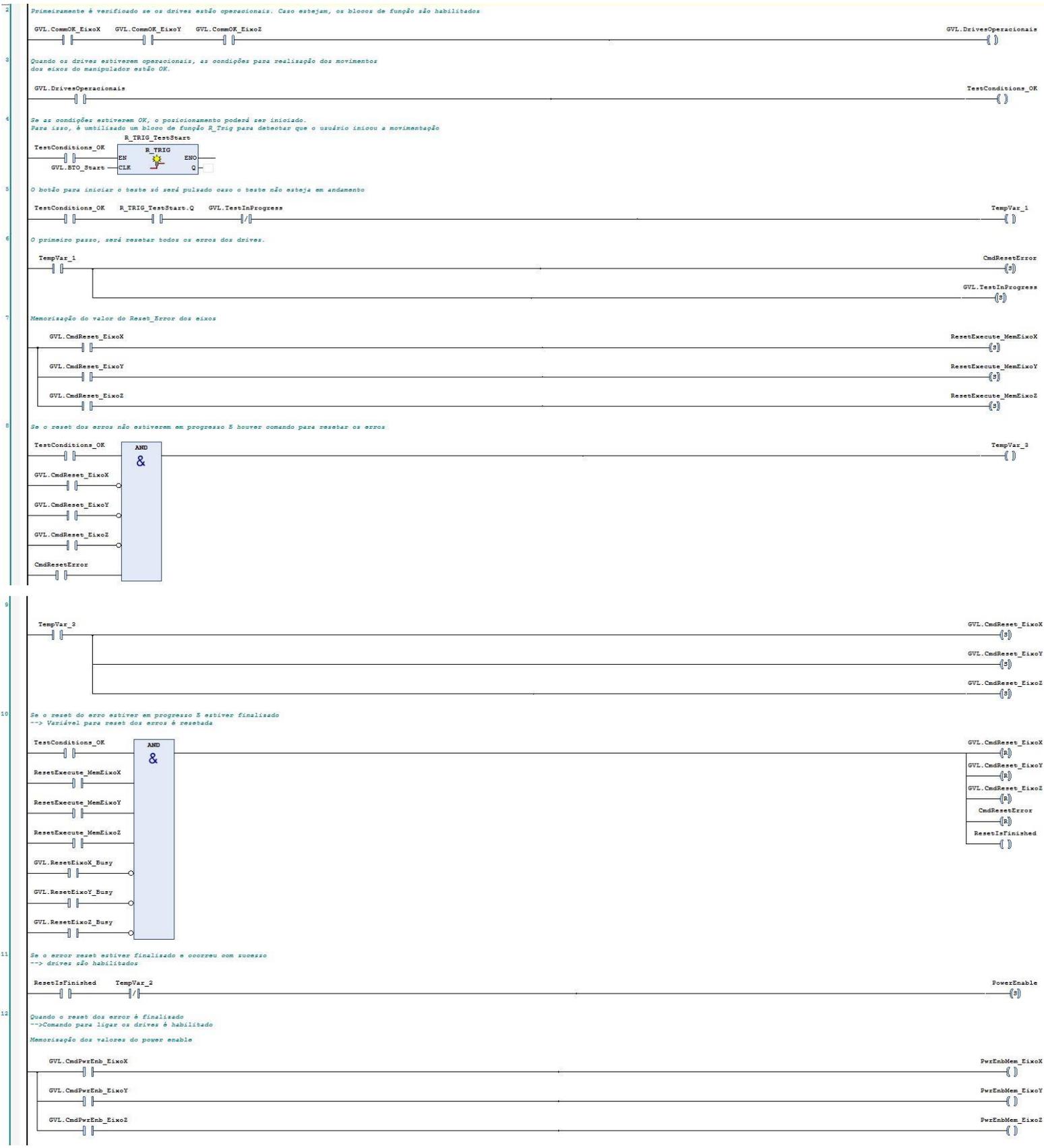
Apêndice I: Programa

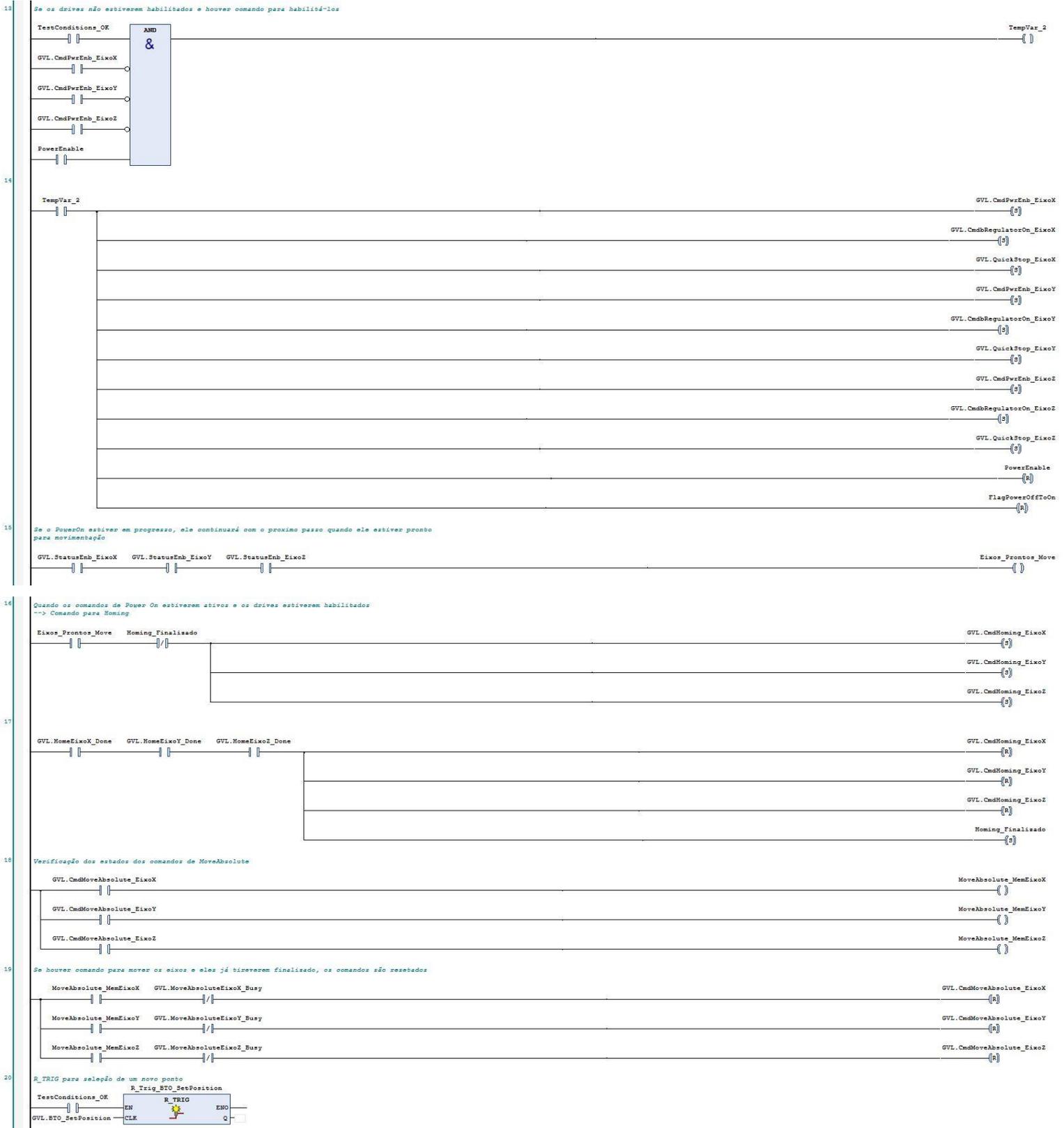


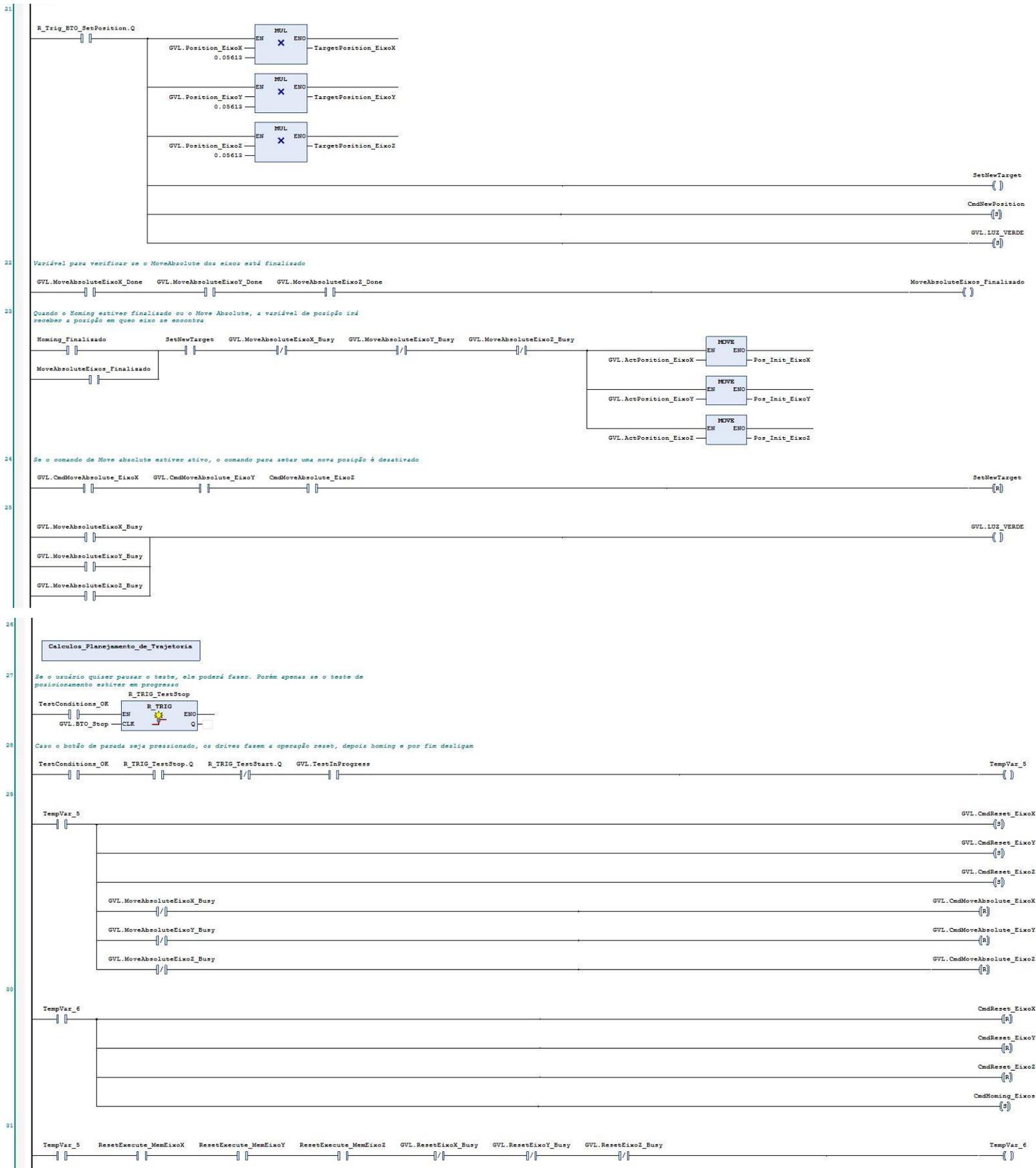




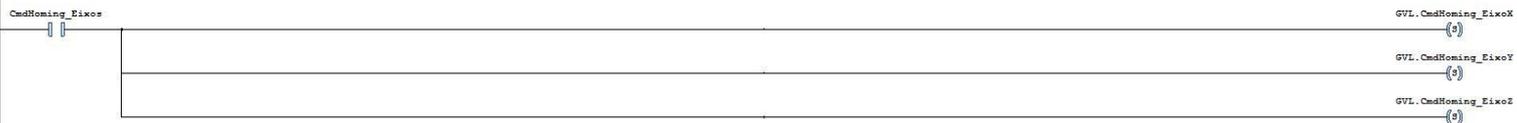






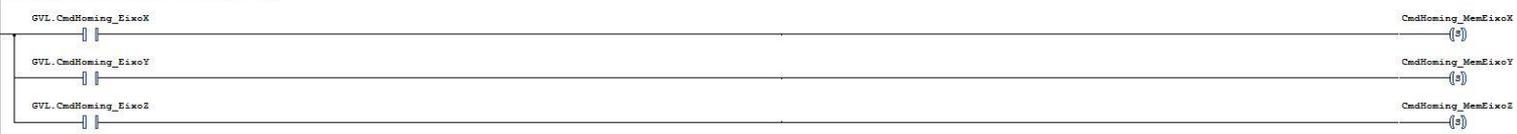


22

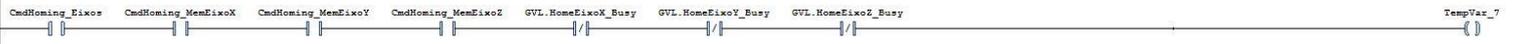


23

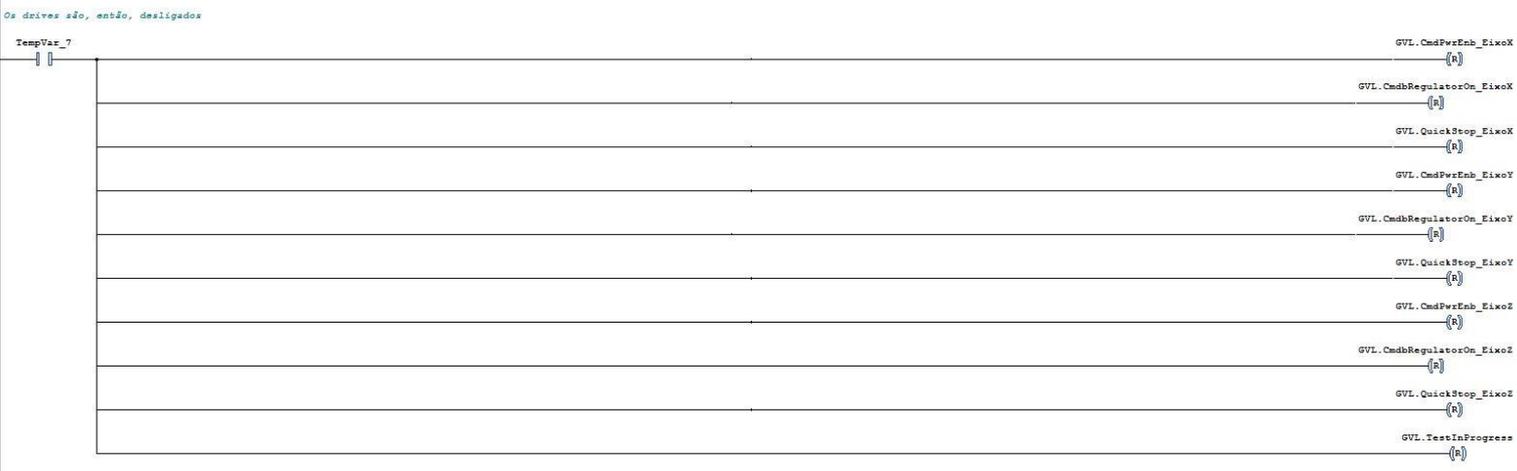
Memorização do estado dos comandos de homing



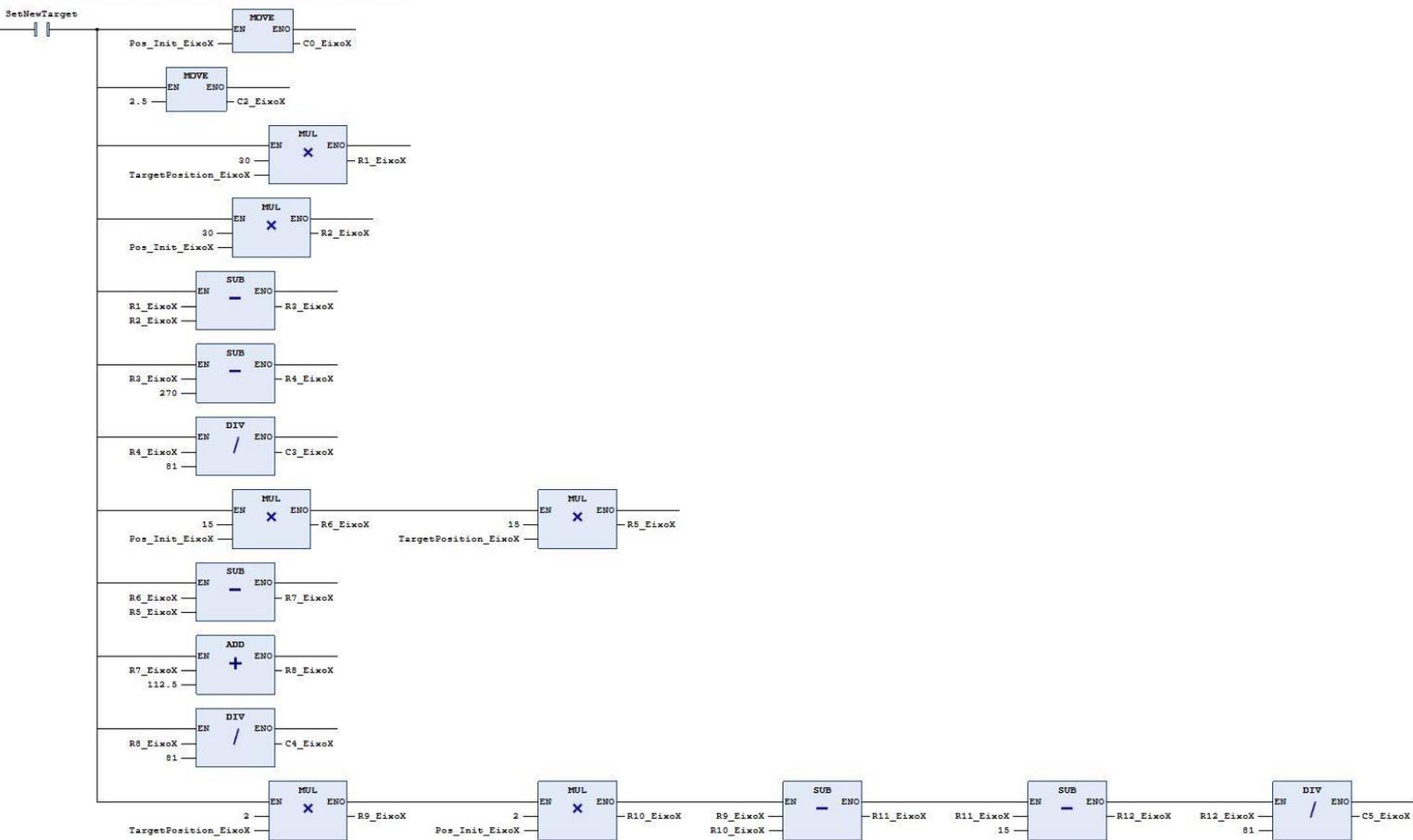
24



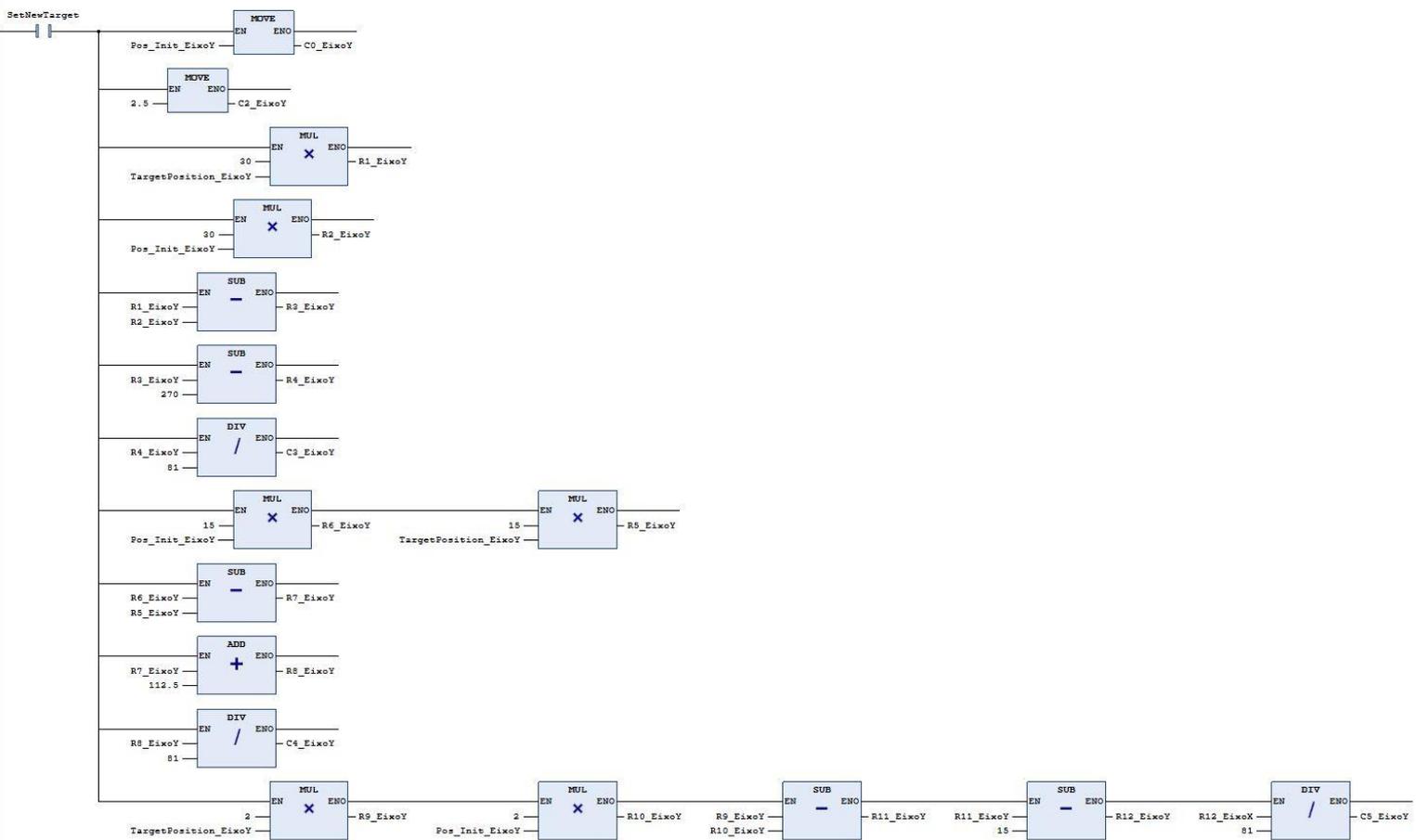
25



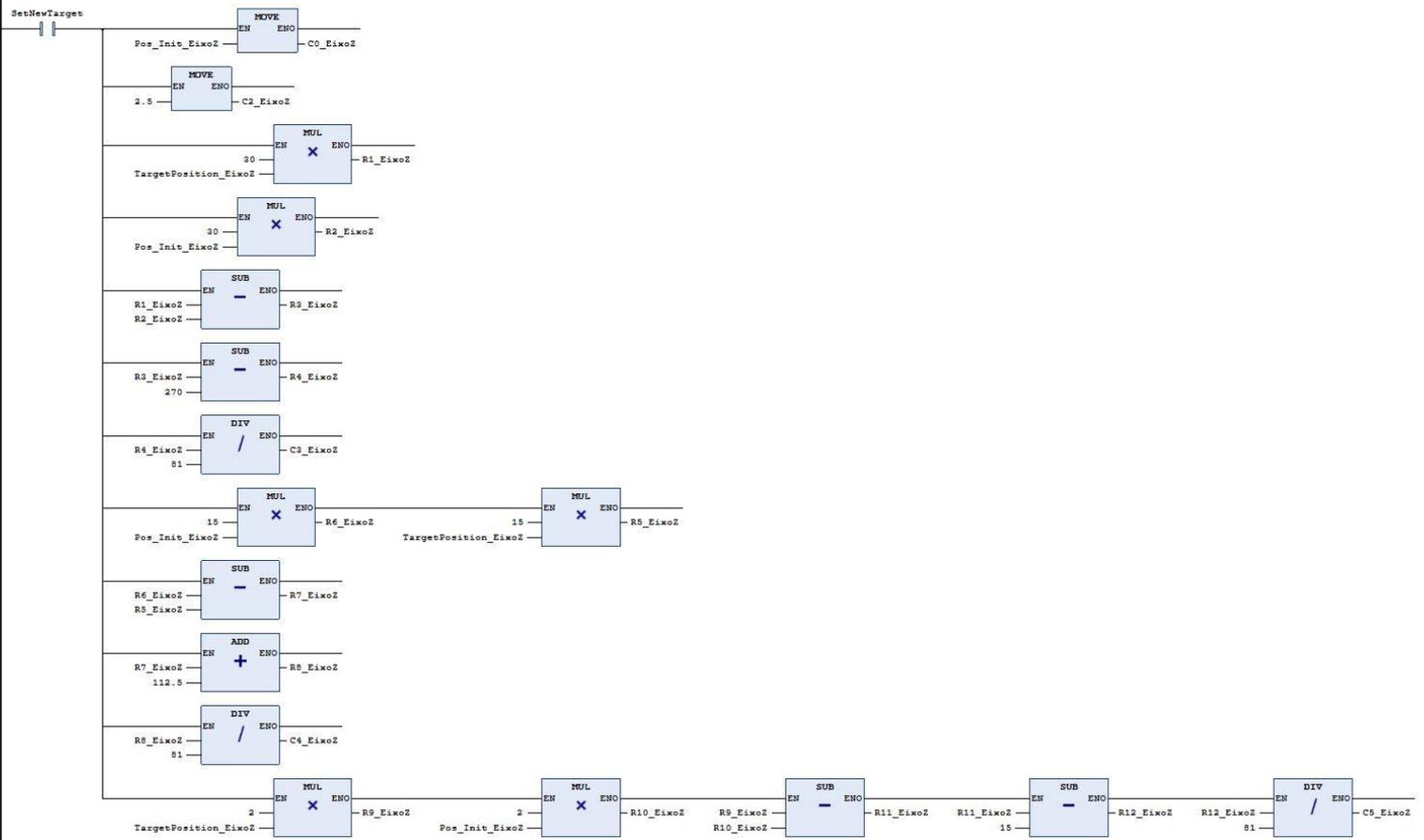
Cálculo das constantes para o planejamento de trajetórias do eixo X



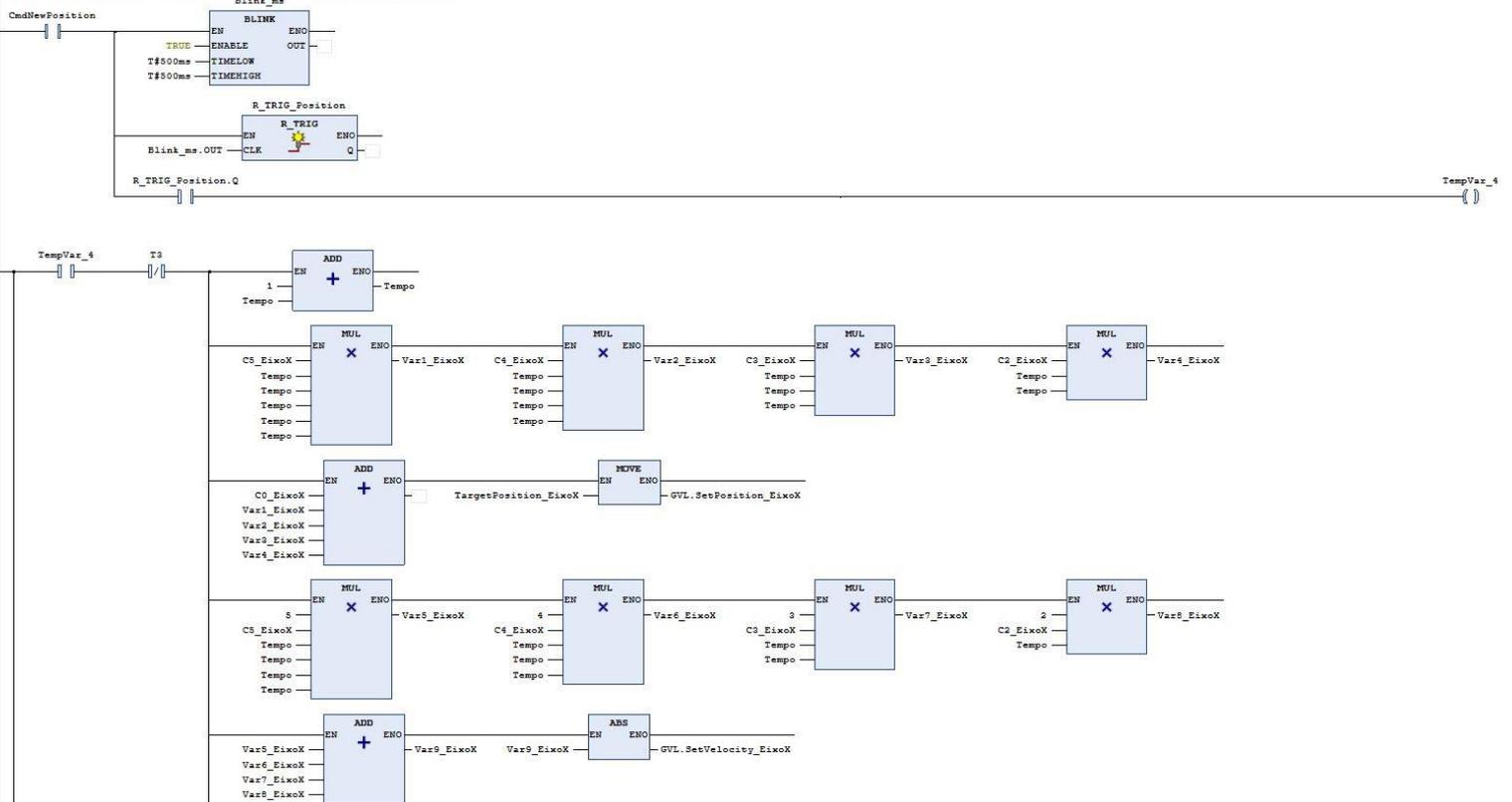
Cálculo das constantes para o planejamento de trajetórias do eixo Y

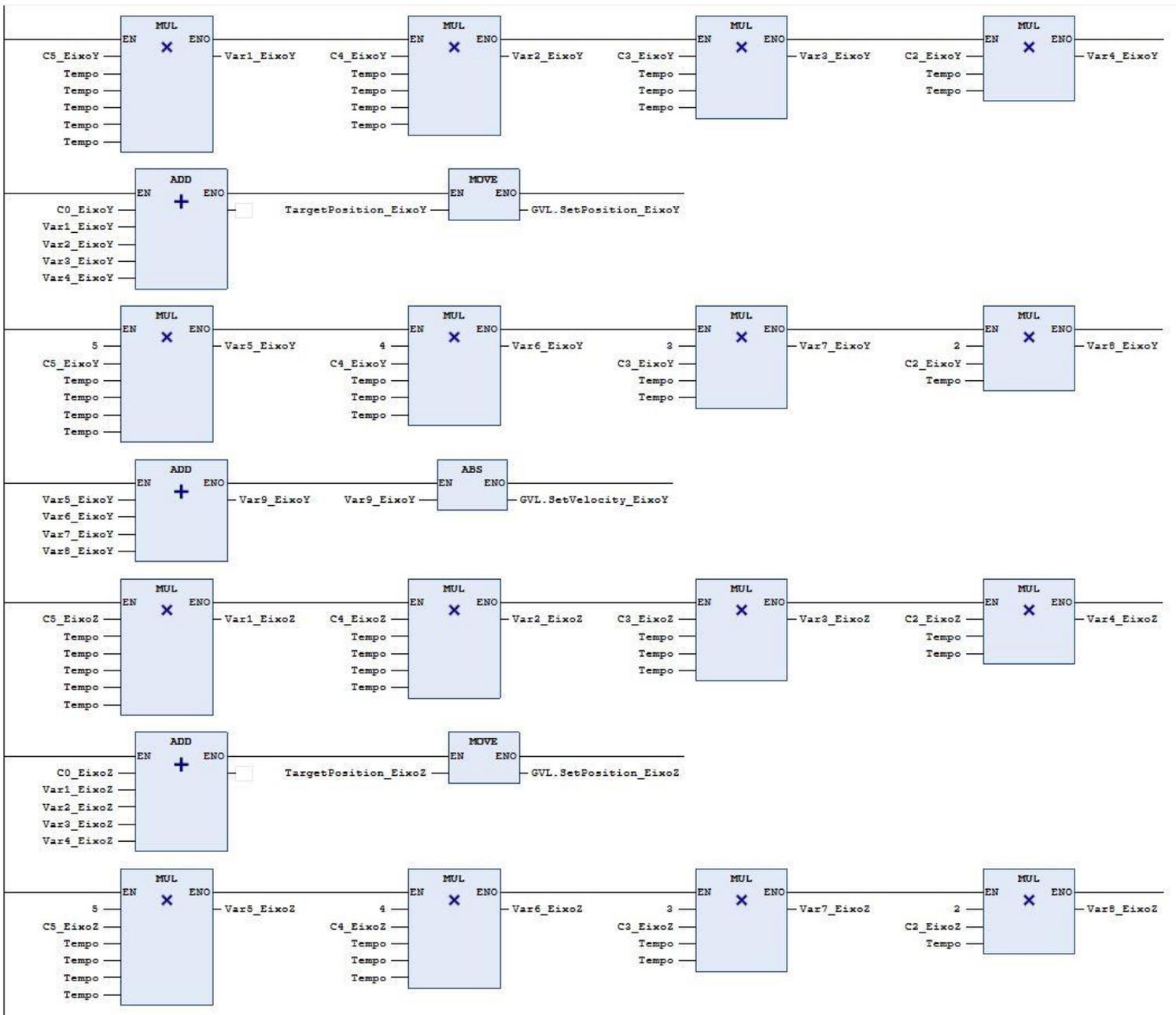


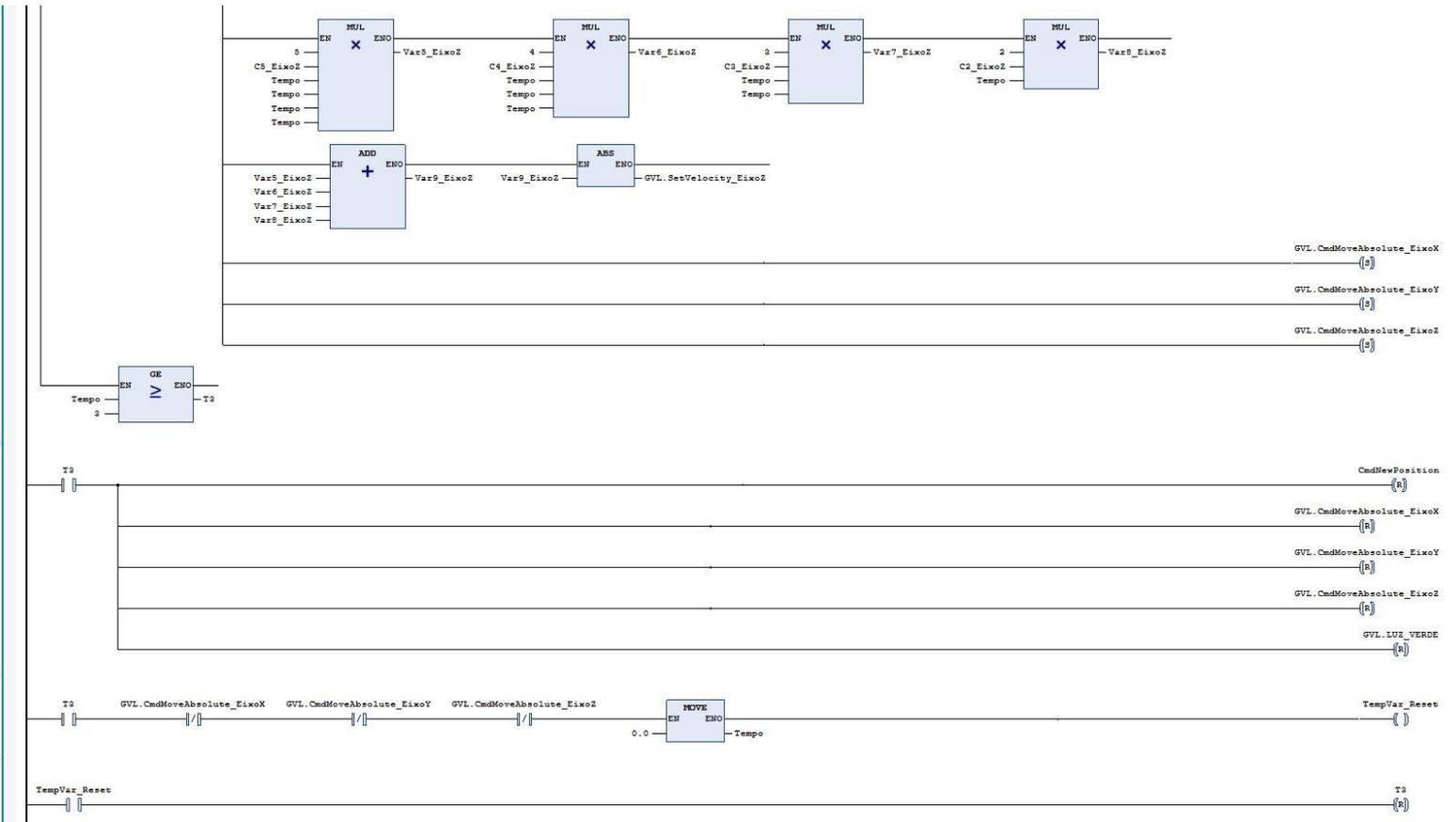
Cálculo das constantes para o planejamento de trajetórias do eixo Z



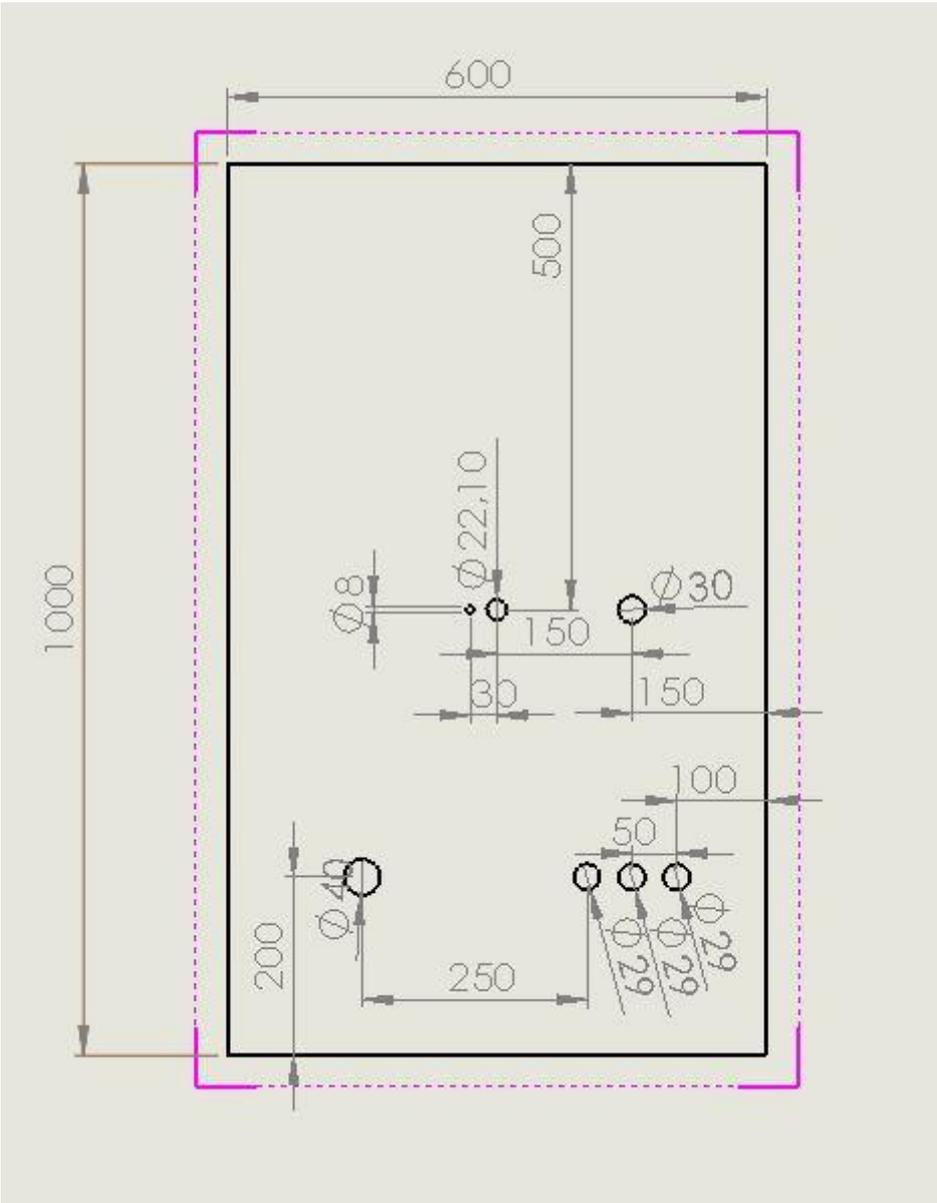
A posição mudará a cada milissegundo, até que a posição alvo seja alcançada



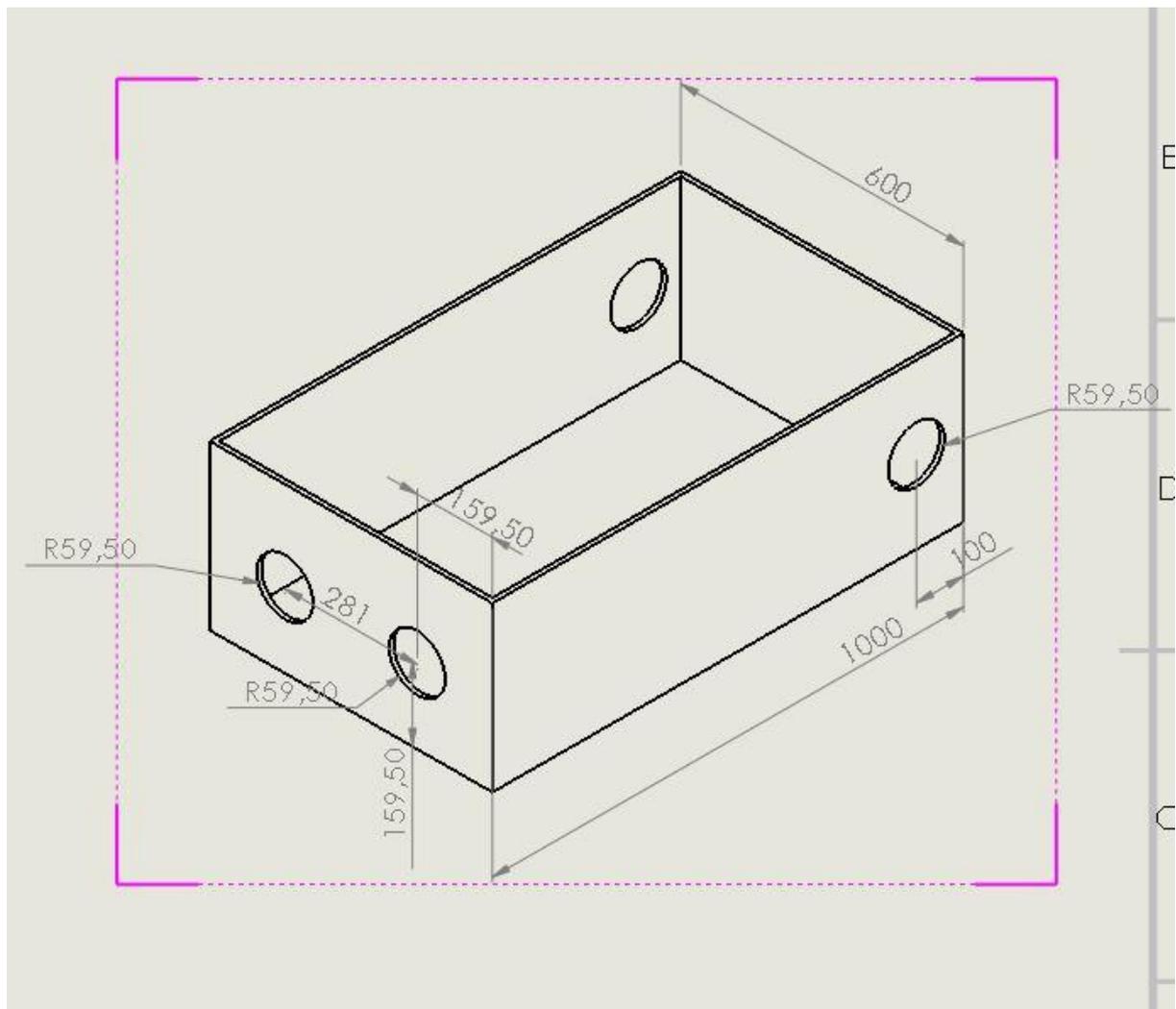




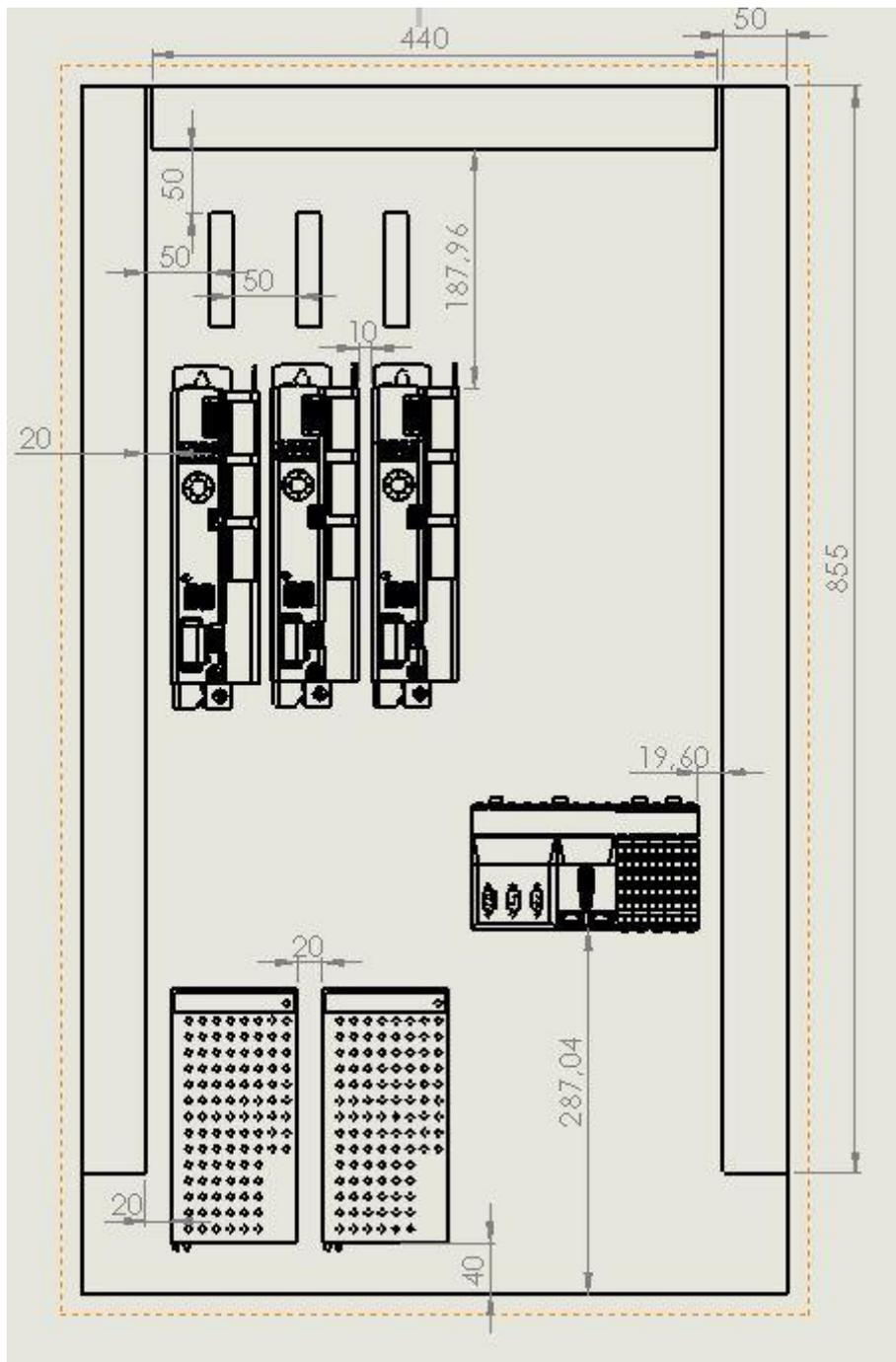
Apêndice II: Desenho técnico da caixa de montagem



Apêndice II.1 - Tampa

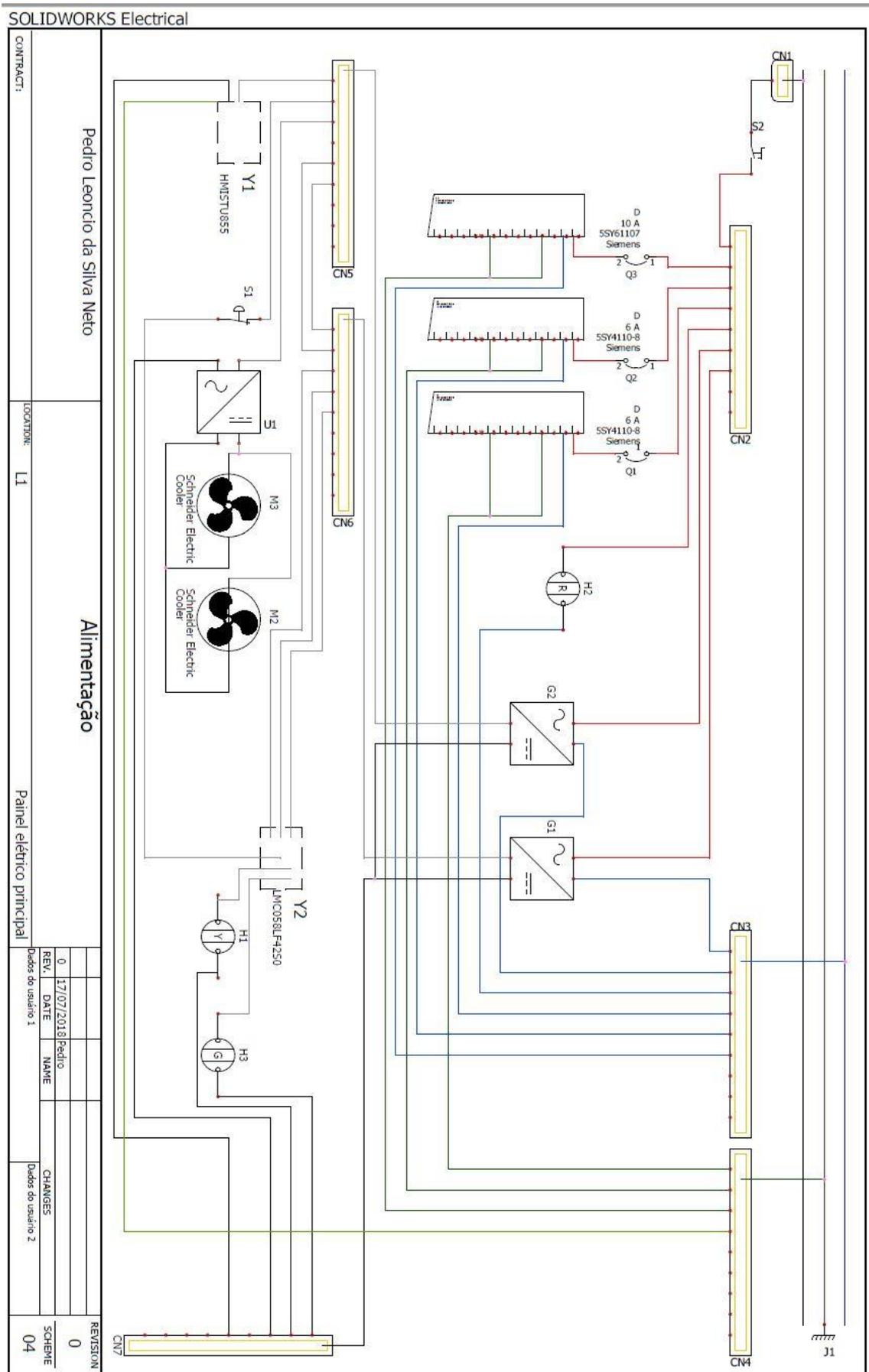


Apêndice II.2 - Caixa



Apêndice II.3 - Arranjo dos dispositivos no painel

Apêndice III: Desenho técnico da instalação elétrica



Apêndice III.1 - Desenho técnico da instalação elétrica

SIGLA	DISPOSITIVO
CN1, CN2, CN3, CN4, CN5, CN6 e CN7	Módulos de borneiras
S1	Botão de emergência
S2	Chave liga-desliga
Q1	Disjuntor de 6A ligado ao servo-drive LXM32MU90M2 do eixo Z
Q2	Disjuntor de 6A ligado ao servo-drive LXM32MU90M2 do eixo Y
Q3	Disjuntor de 10A ligado ao servo-drive LXM32MD18M2 do eixo X
H1	Luz indicadora amarela
H2	Luz indicadora vermelha
H3	Luz indicadora verde
G1 e G2	Fonte de energia de 24V
U1	Dispositivo redutor de tensão
Y1	Interface Homem-Máquina
Y2	Controlador de movimentos
M2 e M3	<i>Coolers</i>
J1	Central de aterramento do painel

Apêndice III.2 - Legenda dos dispositivos utilizados no desenho técnico da instalação elétrica

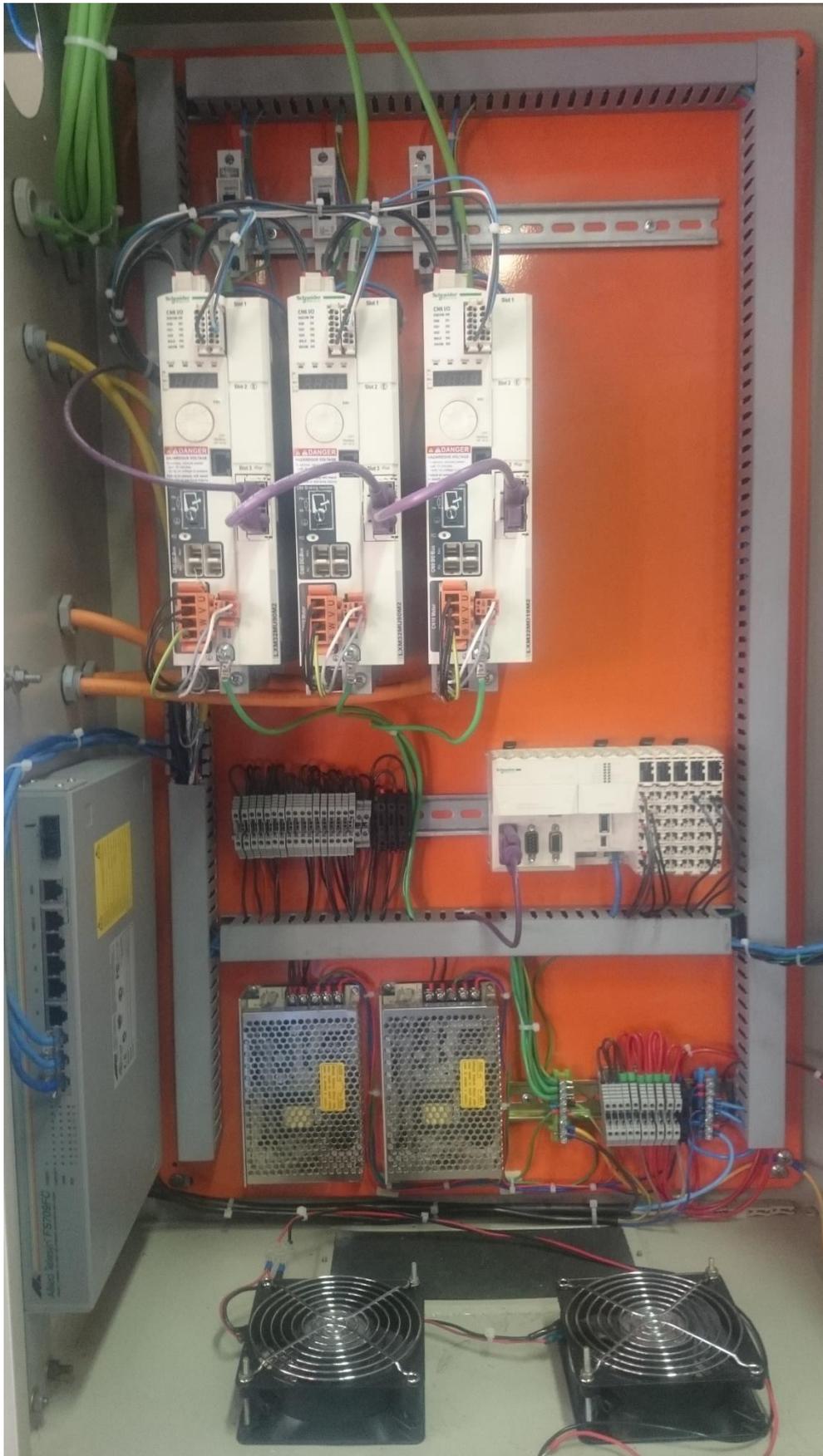
Apêndice IV: Resultado da instalação do robô e dos dispositivos



Apêndice IV.1 – Dispositivos instalados na tampa do painel de controle (vista frontal)



Apêndice IV.2 – Dispositivos instalados na tampa do painel de controle (vista traseira)



Apêndice IV.3 - Dispositivos instalados dentro do painel de controle



Apêndice IV.4 - Vista geral do robô MAXR23-S42-H42-C42



Apêndice IV.5 - Vista dos eixos Y e Z do robô



Apêndice IV.6 - Vista do eixo X do robô